



# Metrics and barycenters for point pattern data

Raoul Müller<sup>1</sup> · Dominic Schuhmacher<sup>1</sup> · Jorge Mateu<sup>2</sup>

Received: 19 September 2019 / Accepted: 11 February 2020 / Published online: 24 February 2020  
© The Author(s) 2020

## Abstract

We introduce the transport–transform and the relative transport–transform metrics between finite point patterns on a general space, which provide a unified framework for earlier point pattern metrics, in particular the generalized spike time and the normalized and unnormalized optimal subpattern assignment metrics. Our main focus is on barycenters, i.e., minimizers of a  $q$ -th-order Fréchet functional with respect to these metrics. We present a heuristic algorithm that terminates in a local minimum and is shown to be fast and reliable in a simulation study. The algorithm serves as a general plug-in method that can be applied to point patterns on any state space where an appropriate algorithm for solving the location problem for individual points is available. We present applications to geocoded data of crimes in Euclidean space and on a street network, illustrating that barycenters serve as informative summary statistics. Our work is a first step toward statistical inference in covariate-based models of repeated point pattern observations.

**Keywords** Fréchet mean · Fréchet median · Network · Optimal transport · Point process · Unbalanced · Wasserstein

## 1 Introduction

Point pattern data are abundant in modern scientific studies. From biomedical imagery over geo-referenced disease cases and positions of mobile phone users to climate change-related space–time events, such as landslides, we have more and more complicated data available. See Chiaraviglio et al. (2016), Lombardo et al. (2018), Konstantinoudis et al. (2019), Samartsidis et al. (2019) for individual examples and the textbooks Diggle (2013), Baddeley et al. (2015), Błaszczyszyn et al. (2018) for a broad overview of further applications. While a few decades ago, data consisted typically of a single point pattern in a low-dimensional Euclidean space, maybe with some low-dimensional mark information, we have nowadays often multiple observations of point patterns available that may live on more complicated spaces,

e.g., manifolds (including shape spaces), spaces of convex sets or function spaces. A setting that has received a particularly large amount of attention recently is point patterns on graphs, such as street networks, see Moradi et al. (2018), Moradi and Mateu (2019) and Rakshit et al. (2019) among others.

Multiple point pattern observations may occur by i.i.d. replication (e.g., of a biological experiment), but may also be governed by one or several covariates or form a time series of possibly dependent patterns. Additional mark information can easily be high-dimensional. Methodology for treating such point pattern data in all these situations is the subject of ongoing statistical research, see, e.g., Baddeley et al. (2015).

From a more abstract point of view, consider the set  $\mathfrak{N}_{\text{fin}}$  of finite counting measures on some metric space  $(\mathcal{X}, d)$ . If we manage to equip  $\mathfrak{N}_{\text{fin}}$  with a metric  $\tau$  that reflects the concept of distance between point patterns in an appropriate problem-related way, there are a number of standard methods which can be applied, including multidimensional scaling, discriminant and cluster analysis techniques. This is a stance already taken in Schuhmacher (2014), Section 1.4, and Mateu et al. (2015). In the metric space  $(\mathfrak{N}_{\text{fin}}, \tau)$ , we can furthermore define a Fréchet mean of order  $q \geq 1$ ; that is, for data  $\xi_1, \dots, \xi_k \in \mathfrak{N}_{\text{fin}}$  any  $\zeta \in \mathfrak{N}_{\text{fin}}$  minimizing

---

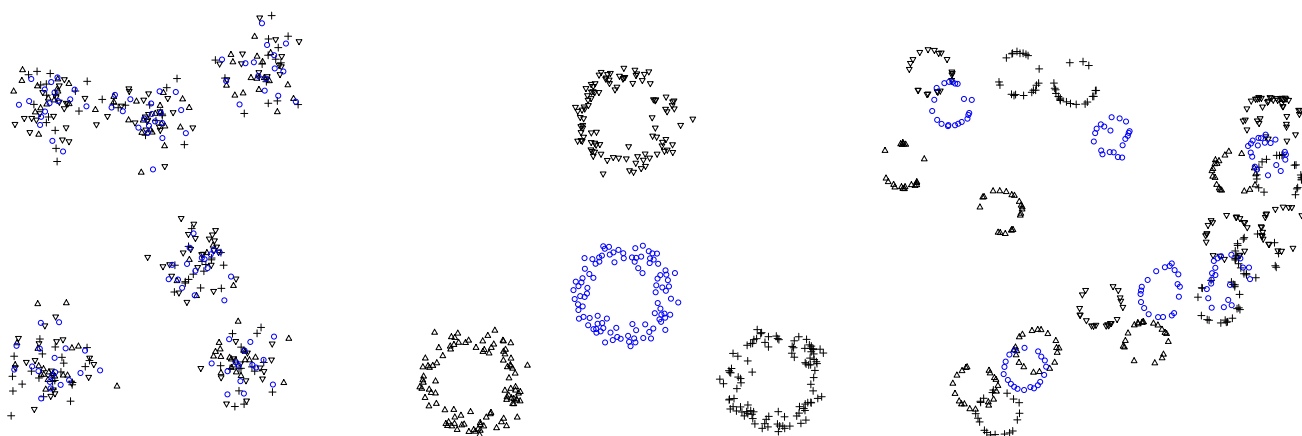
Raoul Müller is supported by *Deutsche Forschungsgemeinschaft* GRK 2088. Jorge Mateu is supported by MTM2016-78917-R from the *Spanish Ministry of Economy and Competitiveness*.

---

✉ Raoul Müller  
raoul.mueller@uni-goettingen.de

<sup>1</sup> Institute for Mathematical Stochastics, University of Göttingen, 37077 Göttingen, Germany

<sup>2</sup> Department of Mathematics, University Jaume I, 12071 Castellón, Spain



**Fig. 1** An example of barycenters computed by our algorithm for three different data sets. In each panel, there are three data point patterns indicated by different symbols (black). The resulting (pseudo-)barycenter

pattern with respect to Euclidean distance is given by the blue circles ( $p = q = 2$ ). (Color figure online)

$$\sum_{j=1}^k \tau(\xi_j, \zeta)^q. \tag{1}$$

Such a  $q$ -th-order mean may serve as a “typical” element of  $\mathfrak{N}_{\text{fin}}$  to represent the data and gives rise to more complex statistical analyses, such as Fréchet regression; see Lin and Müller (2019) and Petersen and Müller (2019).

Two metrics on the space of point patterns that have been widely used are the spike time metric, see Victor and Purpura (1997) for one dimension and Diez et al. (2012) for higher dimension, and the optimal subpattern assignment (OSPA) metric, see Schuhmacher and Xia (2008) and Schuhmacher et al. (2008). In the present paper, we introduce the *transport–transform (TT) metric* and its normalized version, the *relative transport–transform (RTT) metric*, which provide a unified framework for the earlier metrics. Both the TT- and the RTT-metrics are based on matching the points between two point patterns on  $\mathcal{X}$  optimally in terms of some power  $p$  of  $d$  and penalizing points that cannot be reasonably matched. We may interpret these metrics as unbalanced  $p$ -th-order Wasserstein metrics, see Remark 3 below. In the present paper, we always set  $p = q$ .

Among others Schoenberg and Tranbarger (2008), Diez et al. (2012) and Mateu et al. (2015) have treated Fréchet means of order 1 (medians) for the spike time metric under the name of prototypes. However, computations in 2d and higher were only possible for very small data sets due to a prohibitive computational cost of  $O(n^6)$  for the distance between two point patterns with  $n$  points each. In the present work, we use an adapted auction algorithm that is able to compute TT- and RTT-distances between point patterns in  $O(n^3)$ . We further provide a heuristic algorithm that bears some resemblance to a  $k$ -means cluster algorithm and is able to compute local minima of the barycenter problem very efficiently. This makes it possible to compute “quasi-barycenters” for 100 pat-

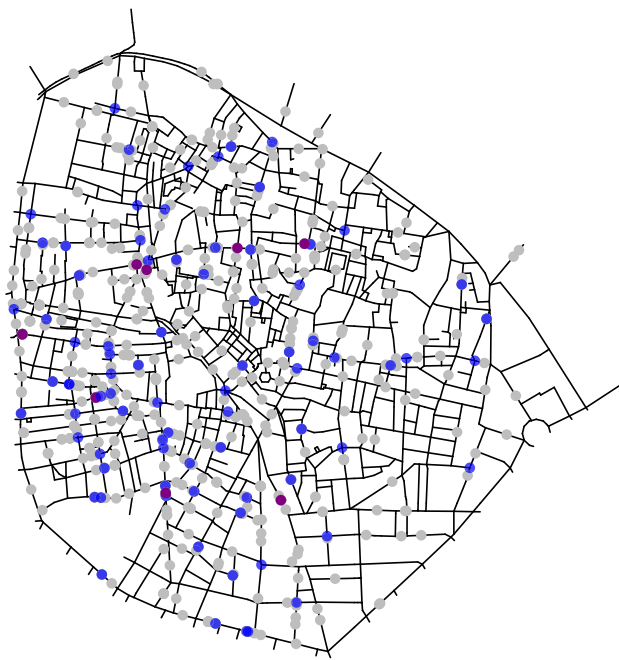
terns of 100 points in  $\mathbb{R}^2$  in a few seconds when basing the TT-distance on the Euclidean distance between points and choosing  $p = q = 2$ .

In Fig. 1, we show some typical barycenters obtained by our algorithm in this setting. We use smaller data sets for better visibility. In each scenario, there are three different point patterns distinguished by the different symbols in black. The (pseudo-)barycenter represented by the blue circles captures the characteristics of each data set rather well. Some minor irregularities, especially in the third panel, may be due to the fact that only a (good) local optimum is computed.

More important than being fast for point pattern data on  $\mathbb{R}^D$  when using squared Euclidean distances is the fact that our algorithm provides a general plug-in method that can in principle be used for point patterns on *any* underlying space  $\mathcal{X}$  where an appropriate “cost function” between objects is specified as  $p$ -th power of a metric  $d$ . All that is required is an algorithm that finds (maybe heuristically) a  $p$ -th-order Fréchet mean for individual points in  $\mathcal{X}$ , i.e., finds  $z \in \mathcal{X}$  minimizing  $\sum_{j=1}^k d(x_j, z)^p$  for any given  $x_1, \dots, x_k \in \mathcal{X}$ . We refer to this in what follows as the underlying *location problem*. The reduction to the underlying location problem allows us to treat the case of point patterns on a network equipped with the shortest-path metric and  $p = 1$ . Figure 2 gives an example for crime data in Valencia, Spain, which we study in more detail in Sect. 6.

The barycenter problem we consider in this paper is closely related to the problem of computing an unbalanced Wasserstein barycenter, see, e.g., Chizat et al. (2018). However, rather than minimizing a Fréchet functional on the space of all measures, we minimize on the space  $\mathfrak{N}_{\text{fin}}$  of  $\mathbb{Z}_+$ -valued measures, see Remark 5.

The plan of the paper is as follows. In Sect. 2, we introduce the TT- and RTT-metrics and discuss their relations to spike time, OSPA, and incomplete Wasserstein metrics. Sec-



**Fig. 2** An example of a barycenter on a street network. Shown are 8 patterns of assault crimes during the summer months of 2010–2017 in the old town of Valencia (all in gray for better overall visibility). The resulting barycenter with respect to shortest-path distance along the streets is given in blue, with multipoints in purple ( $p = q = 1$ ). (Color figure online)

tion 3 specifies what we mean by a barycenter (or Fréchet mean) with respect to these metrics and gives an important result that forms the basis for our heuristic algorithm. Two versions of this algorithm, a more direct one and an improved one, which saves computation steps that are unlikely to substantially influence the final result, are discussed in detail in Sect. 4, along with some practical aspects. Section 5 contains a larger simulation study, which investigates robustness and runtime performances of the two algorithms for the case of Euclidean distance and  $p = 2$ . Finally, we give two applications to data of crime events on a city map for real data in Sect. 6. The first one concerns street thefts in Bogotá, Colombia. We treat this again as data in Euclidean space, using  $p = 2$ . The second one deals with assault cases in the streets of Valencia, Spain. Here, we compute barycenters based on the actual shortest-path distance on the street network and use  $p = 1$ .

## 2 The transport–transform metric

Denote by  $\mathfrak{N}_{\text{fin}}$  the space of finite point patterns (counting measures) on a complete separable metric space  $(\mathcal{X}, d)$ , equipped with the usual  $\sigma$ -algebra  $\mathcal{N}_{\text{fin}}$  generated by the point count maps  $\Psi_A: \mathfrak{N}_{\text{fin}} \rightarrow \mathbb{R}, \xi \mapsto \xi(A)$  for  $A \subset \mathcal{X}$  Borel measurable. Elements of  $\mathfrak{N}_{\text{fin}}$  are typically denoted by  $\xi, \eta, \zeta$  here. As usual, we write  $\delta_x$  for the Dirac measure with unit

mass at  $x \in \mathcal{X}$ . In the present section, we mostly use measure notation such as  $\xi = \sum_{i=1}^n \delta_{x_i}, \xi(\{x\}) \geq 1$  or  $\xi + \eta$ , but in later sections we also use corresponding (multi)set notation such as  $\xi = \{x_1, \dots, x_n\}, x \in \xi$  or  $\xi \cup \eta$  where this is unambiguous.

We use  $|\xi| = \xi(\mathcal{X})$  to denote the total number of points in the pattern  $\xi$ . For  $n \in \mathbb{Z}_+ = \{0, 1, 2, \dots\}$  write  $[n] = \{1, 2, \dots, n\}$  (including  $[0] = \emptyset$ ) and denote by  $\mathfrak{N}_n$  the set of point patterns with exactly  $n$  points. We first introduce the metrics we use on  $\mathfrak{N}_{\text{fin}}$ , which unify and generalize two of the main metrics used previously in the literature.

**Definition 1** Let  $C > 0$  and  $p \geq 1$  be two parameters, referred to as *penalty* and *order*, respectively.

- (a) For  $\xi = \sum_{i=1}^m \delta_{x_i}, \eta = \sum_{j=1}^n \delta_{y_j} \in \mathfrak{N}_{\text{fin}}$ , define the *transport–transform (TT) metric* by

$$\begin{aligned} \tau(\xi, \eta) &= \tau_{C,p}(\xi, \eta) \\ &= \left( \min \left( (m+n-2l)C^p + \sum_{r=1}^l d(x_{i_r}, y_{j_r})^p \right) \right)^{1/p}, \end{aligned} \tag{2}$$

where the minimum is taken over equal numbers of pairwise different indices  $i_1, \dots, i_l$  in  $[m]$  and  $j_1, \dots, j_l$  in  $[n]$ , i.e. over the set

$$\begin{aligned} S(m, n) &= \{(i_1, \dots, i_l; j_1, \dots, j_l); \\ & \quad l \in \{0, 1, \dots, \min\{m, n\}\}, \\ & \quad i_1, \dots, i_l \in [m] \text{ pairwise different,} \\ & \quad j_1, \dots, j_l \in [n] \text{ pairwise different}\}. \end{aligned}$$

- (b) For  $\xi, \eta \in \mathfrak{N}_{\text{fin}}$ , define the *relative transport–transform (RTT) metric* by

$$\bar{\tau}(\xi, \eta) = \bar{\tau}_{C,p}(\xi, \eta) = \frac{1}{\max\{|\xi|, |\eta|\}^{1/p}} \tau_{C,p}(\xi, \eta). \tag{3}$$

We state and prove below that  $\tau$  and  $\bar{\tau}$  are indeed metrics.

The following result simplifies proofs of statements about these metrics and is furthermore invaluable for their computation. The idea is to extend the metric space  $(\mathcal{X}, d \wedge (2^{1/p}C))$ , where  $[d \wedge (2^{1/p}C)](x, y) = \min\{d(x, y), 2^{1/p}C\}$ , by setting  $\mathcal{X}' = \mathcal{X} \cup \{\mathfrak{N}\}$  for an auxiliary element  $\mathfrak{N} \notin \mathcal{X}$  and

$$d'(x, y) = \begin{cases} \min\{d(x, y), 2^{1/p}C\} & \text{if } x, y \in \mathcal{X}; \\ C & \text{if } \mathfrak{N} \in \{x, y\}, x \neq y; \\ 0 & \text{if } x = y = \mathfrak{N}. \end{cases}$$

It is shown in Lemma A.1 that  $(\mathcal{X}', d')$  is a metric space again. We may then compute distances in the  $\tau$  and  $\bar{\tau}$  metrics by

solving an optimal matching problem between point patterns with the same cardinality. For  $n \in \mathbb{N}$  denote by  $S_n$ , the set of permutations on  $[n]$ .

**Theorem 1** *Let  $\xi = \sum_{i=1}^m \delta_{x_i}, \eta = \sum_{j=1}^n \delta_{y_j} \in \mathfrak{M}_{\text{fin}}$ , where w.l.o.g.  $m \leq n$  (otherwise swap  $\xi$  and  $\eta$ ). Set  $x_i = \aleph$  for  $m + 1 \leq i \leq n$  and  $\tilde{\xi} = \sum_{i=1}^n \delta_{x_i}$ . Then,*

$$\tau(\xi, \eta) = \left( \min_{\pi \in S_n} \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p \right)^{1/p} \quad \text{and}$$

$$\bar{\tau}(\xi, \eta) = \left( \frac{1}{n} \min_{\pi \in S_n} \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p \right)^{1/p}.$$

The proof of this and the other theorems in this section can be found in the appendix.

**Remark 1** (Computation of TT- and RTT-metrics) Writing  $n$  for the maximum cardinality as in Theorem 1, this result shows that we can compute both  $\tau(\xi, \eta)$  and  $\bar{\tau}(\xi, \eta)$  in worst-time complexity of  $O(n^3)$  by using the classic Hungarian method for the assignment problem; see Kuhn (1955). In practice, we use the auction algorithm proposed in Bertsekas (1988), because it has usually much better runtime in our experience, although the default version has a somewhat worse worst-case performance of  $O(n^3 \log(n))$ .<sup>1</sup>

**Theorem 2** *The maps  $\tau$  and  $\bar{\tau}$  are metrics on  $\mathfrak{M}_{\text{fin}}$ .*

The next result establishes our previous claim that the new transport–transform construction generalizes two metrics on  $\mathfrak{M}_{\text{fin}}$  previously used in the literature.

**Theorem 3**

(a) *If  $p = 1$ , then for any  $\xi, \eta \in \mathfrak{M}_{\text{fin}}$*

$$\tau(\xi, \eta) = \min_{(\xi_0, \dots, \xi_N)} \sum_{i=0}^{N-1} c_{\text{elem}}(\xi_i, \xi_{i+1}), \quad (4)$$

where the minimum is taken over all  $N \in \mathbb{N}$  and all paths  $(\xi_0, \dots, \xi_N) \in \mathfrak{M}_{\text{fin}}^{N+1}$  such that  $\xi_0 = \xi, \xi_N = \eta$ , and from  $\xi_i$  to  $\xi_{i+1}$  either a single point is added or deleted at cost  $c_{\text{elem}}(\xi_i, \xi_{i+1}) = C$  or a single point is moved from  $x$  to  $y$  at cost  $c_{\text{elem}}(\xi_i, \xi_{i+1}) = d(x, y)$ .

(b) *If  $\text{diam}(\mathcal{X}) = \sup_{x, y \in \mathcal{X}} d(x, y) \leq 2^{1/p} C$ , then for any  $\xi = \sum_{i=1}^m \delta_{x_i}, \eta = \sum_{j=1}^n \delta_{y_j} \in \mathfrak{M}_{\text{fin}}$ , assuming w.l.o.g.  $m \leq n$*

$$\bar{\tau}(\xi, \eta)^p = \frac{1}{n} \left( (n - m)C^p + \min_{\pi \in S_n} \sum_{i=1}^m d(x_i, y_{\pi(i)})^p \right). \quad (5)$$

Theorem 3(a) implies that the TT-metric is the same as the spike time metric (using add and delete penalties  $P_a = P_d = C$  and a move penalty  $P_m = 1$ ), which was originally introduced on  $\mathbb{R}_+$  by Victor and Purpura (1997) and generalized to metric spaces by Diez et al. (2012). It can be seen from the proof in the appendix that the right hand side of (4) is not a metric in general if  $p > 1$ .

Theorem 3(b) implies that the RTT-metric is the same as the OSPA metric, introduced in Schuhmacher and Xia (2008) and Schuhmacher et al. (2008). Note that in the definition of the OSPA metric  $\text{diam}(\mathcal{X}) \leq C \leq 2^{1/p} C$  was either required or enforced by taking the minimum of  $d$  with  $C$ . Here, it can be seen that the right hand side of (5) is not a metric in general if  $\text{diam}(\mathcal{X}) > 2C$ .

**Remark 2** (Computation of spike time distances) The spike time distances in Victor and Purpura (1997) and Diez et al. (2012) allowed for separate add and delete penalties  $P_a$  and  $P_d$ , as well as a move penalty  $P_m$  (factor in front of  $d(x, y)$ ). We set here  $P_a = P_d = C$  to obtain a proper metric and divide distances by  $P_m$ , which is just a scaling. Thus, the parameter  $C = P_a/P_m = P_d/P_m$  is all that remains.

As noted at the end of Section 4 in Diez et al. (2012), having different add and delete penalties may be useful for controlling the total number of points in a barycenter point pattern. Let us point out therefore that Theorem 1 is easily adapted to this more general situation by setting  $d'(x, y) = \min\{d(x, y), 2^{1/p}(P_a + P_d)\}$ ,  $d'(\aleph, y) = P_a$  and  $d'(x, \aleph) = P_d$  for all  $x, y \in \mathcal{X}$ .

In particular, this yields a worst-time complexity of  $O(n^3)$  for general (maybe asymmetric) spike time distances in general metric spaces, which is a substantial improvement over the  $O(n^6)$  complexity of the incremental matching algorithm presented in Diez et al. (2012).

**Remark 3** (Unbalanced Wasserstein metrics) The TT- and RTT-metrics can be seen as unbalanced Wasserstein metrics, see, e.g., Chizat et al. (2018), Liero et al. (2018) and the references therein. Minimizing over the space  $\mathfrak{M}_{\text{fin}}$  of all finite measures on  $\mathcal{X} \times \mathcal{X}$ , we obtain the TT-distance as a solution to a particular instance of the unbalanced optimal transport problem in Chizat et al. (2018), Definition 2.11, namely

$$\tau(\xi, \eta)^p = \inf_{\gamma \in \mathfrak{M}_{\text{fin}}} \left( \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p \gamma(dx, dy) + C^p \|\xi - \gamma_1\|_{\text{TV}} + C^p \|\eta - \gamma_2\|_{\text{TV}} \right), \quad (6)$$

<sup>1</sup> There is a modified auction algorithm that can improve the worst-case performance to  $O(n^{5/2} \log(n))$ ; for the performance discussion see Bertsekas (1988), page 109. Actually, both orders include a factor  $c$  in the log which measures the numerical precision, assumed to be bounded here.

where  $\gamma_1 = \gamma(\cdot \times \mathcal{X})$  and  $\gamma_2 = \gamma(\mathcal{X} \times \cdot)$  denote the marginals of  $\gamma$ , and  $\|\cdot\|_{TV}$  is the total variation norm of signed measures; specifically  $\|\mu - \nu\|_{TV} = \sup_A (\mu(A) - \nu(A)) + \sup_{A^c} (\nu(A) - \mu(A))$  for  $\mu, \nu \in \mathfrak{M}_{fin}$ , where the suprema are taken over all measurable subsets of  $\mathcal{X}$ .

Equation (6) can be shown as follows. It is straightforward to see that we may take the infimum on the right hand side only over  $\gamma \in \mathfrak{M}_{fin}$  with marginals  $\gamma_1 \leq \xi$  and  $\gamma_2 \leq \eta$ , because any additional mass in  $\gamma$  may be removed without increasing the total cost of  $\gamma$ . Writing  $\xi = \sum_{i=1}^n \delta_{x_i}$  and  $\eta = \sum_{i=1}^n \delta_{y_i}$  with the help of additional points at  $\mathfrak{N}$  (if necessary), we obtain by similar arguments as in the proof of Theorem 1 that the latter problem is equivalent to the discrete transportation problem

$$\begin{aligned} \min_{(\gamma_{ij})_{1 \leq i, j \leq n}} & \sum_{i, j=1}^n d'(x_i, y_j)^p \cdot \gamma_{ij} \\ \text{s.t.} & \sum_{j=1}^n \gamma_{ij} = 1 \text{ for all } i, \quad \sum_{i=1}^n \gamma_{ij} = 1 \text{ for all } j, \\ & \gamma_{ij} \geq 0 \text{ for all } i, j. \end{aligned}$$

It is a standard result in linear programming that this problem always has a solution  $\gamma_{ij} \in \{0, 1\}$ ,  $1 \leq i, j \leq n$ ; see, e.g., the theorem in Section 6.5 of Luenberger and Ye (2008), which is essentially due to the fact that the structure of the constraint allows for a back substitution approach involving only additions and subtractions. We may therefore conclude from Theorem 1 that Equation (6) holds and that the infimum on the right hand side is attained for  $\gamma = \sum_{i, j=1}^n \mathbb{1}\{x_i, y_j \neq \mathfrak{N}\} \gamma_{ij} \delta_{(x_i, y_j)}$ .

In principle, Remark 3 allows us to specialize results and algorithms for unbalanced Wasserstein metrics to TT- and RTT-metrics. However, the discrete setting we consider here is sometimes not included in the general theorems or requires a more specialized treatment. Algorithms for computing unbalanced transport plans are typically derived from balanced optimal transport algorithms; a selection can be found in Chizat (2017). The auction algorithm we use in this paper is derived from the auction algorithm used for balanced assignment problems in a similar way.

### 3 Barycenters with respect to the TT-metric

For data on quite general metric spaces, barycenters can formalize the idea of a center element representing the data. In the case of  $\mathfrak{N}_{fin}$ , we are thus looking for a center point pattern that gives a good first-order representation of a set of data point patterns  $\xi_1, \dots, \xi_k$ . More formally, we may define a barycenter as the (weighted)  $q$ -th-order Fréchet mean with respect to  $\tau$ ; see Fréchet (1948).

**Definition 2** For  $k \in \mathbb{N}$ , let  $\xi_1, \dots, \xi_k \in \mathfrak{N}_{fin}$  be data point patterns and  $\lambda_1, \dots, \lambda_k > 0$  with  $\sum_{j=1}^k \lambda_j = 1$  be weights. Let furthermore  $q \geq 1$ . Then, we call any

$$\zeta_* \in \arg \min_{\zeta \in \mathfrak{N}_{fin}} \sum_{j=1}^k \lambda_j \tau(\xi_j, \zeta)^q \tag{7}$$

a (weighted) barycenter of order  $q$ . If no weights are specified, we tacitly assume that  $\lambda_j = 1/k$  for  $1 \leq j \leq k$ , leading to an “unweighted” barycenter.

**Remark 4** For  $q = 2$ , barycenters on general metric spaces are simply known as (empirical) Fréchet means. For  $q = 1$ , they are sometimes known as Fréchet medians. This comes from the fact that given  $x_1, \dots, x_k \in \mathbb{R}^D$ , we have

$$\arg \min_{z \in \mathbb{R}^D} \sum_{j=1}^k \|x_j - z\|^2 = \frac{1}{k} \sum_{j=1}^k x_j \tag{8}$$

(the arg min is unique here), and that given  $x_1, \dots, x_k \in \mathbb{R}$ , we have

$$\arg \min_{z \in \mathbb{R}} \sum_{j=1}^k \|x_j - z\| = \text{median}\{x_1, \dots, x_k\}, \tag{9}$$

where the right hand side denotes the set of medians  $\{z \in \mathbb{R}; \#\{j; x_j \leq z\} = \#\{j; x_j \geq z\}\}$ .

**Remark 5** As seen in Remark 3, we may interpret  $\tau$  as an unbalanced Wasserstein metric. There has been a great deal of research on Wasserstein barycenters (in the Fréchet mean sense as above, see, e.g., Agueh and Carlier (2011) or Cuturi and Doucet (2014)), which more recently also extends to unbalanced Wasserstein metrics, see, e.g., Chizat et al. (2018) or Schmitz et al. (2018). In addition to the fact that much of the corresponding theory is not well adapted to the case of discrete input measures, with the notable exception of Anderes et al. (2016), we point out that a fundamental difference of (7) lies in the fact that we minimize over the space  $\mathfrak{N}_{fin}$  of  $\mathbb{Z}_+$ -valued measures. This space is smaller than the space  $\mathfrak{M}_{fin}$  of general finite measures, but has a more complicated structure because it decays into connected components  $\mathfrak{N}_n = \{\xi \in \mathfrak{N}_{fin}; |\xi| = n\}$  (under the TT-metric), implying, e.g., that continuous optimization procedures will not work directly.

In what follows, we always set  $p = q$  and choose this number mostly in  $\{1, 2\}$ . We refer to the resulting barycenters simply as 1- and 2-barycenter or as point pattern median and point pattern mean, respectively. Point pattern medians have been introduced under the name of prototypes in Schoenberg

and Tranbarger (2008) on  $\mathbb{R}$  and studied in higher dimensions in Diez et al. (2012) and Mateu et al. (2015). However, in these papers the applicability was limited to rather small data sets due to the large computation cost of  $O(n^6)$  mentioned in Remark 2.

Using the construction from Theorem 1, we may reformulate the barycenter problem as a multidimensional assignment problem, generalizing Lemma 16 in Koliander et al. (2018). Note that for the TT-metric we can add an arbitrary number of points at  $\aleph$  to both point patterns without changing the minimum in Theorem 1.

**Theorem 4** For point patterns  $\xi_j = \sum_{i=1}^{n_j} \delta_{x_{ij}}$ ,  $j \in [k]$ , let  $\tilde{n} := \lfloor \frac{2}{k+1} \sum_{j=1}^k n_j \rfloor$  and  $n \geq \max\{\tilde{n}, n_j; 1 \leq j \leq k\}$ . Set  $x_{ij} \in \aleph$  for  $n_j + 1 \leq i \leq n$  and  $\tilde{\xi}_j = \sum_{i=1}^n \delta_{x_{ij}}$  for any  $j \in [k]$ .

Then, for any  $\pi_{*,1}, \dots, \pi_{*,k} \in S_n$  jointly minimizing

$$\sum_{i=1}^n \min_{z \in \mathcal{X}'} \sum_{j=1}^k d'(x_{\pi_j(i),j}, z)^p \tag{10}$$

the point pattern  $\zeta_* | \mathcal{X}$  with  $\zeta_* = \sum_{i=1}^n \delta_{z_i}$ , where  $z_i \in \arg \min_{z \in \mathcal{X}'} \sum_{j=1}^k d'(x_{\pi_{*,j}(i),j}, z)^p$  is a  $p$ -th-order barycenter with respect to the TT-metric.

The  $\pi_{*,1}, \dots, \pi_{*,k} \in S_n$  above define  $n$  disjoint “clusters”  $\mathcal{C}_i = \{x_{\pi_{*,j}(i),j}; 1 \leq j \leq k\}$ , where each contains exactly one (maybe virtual) point of each point pattern. The minimization of (10) may thus be interpreted as a multidimensional assignment problem with cluster cost

$$\text{cost}_*(\mathcal{C}) = \min_{z \in \mathcal{X}'} \sum_{x \in \mathcal{C}} d'(x, z)^p. \tag{11}$$

**Proof** Let us first give an upper bound on the cardinality of the barycenter. A single barycenter point can be matched with up to  $k$  points (one from each point pattern). If said point is matched with only  $\frac{k}{2}$  points or fewer, it cannot be worse to delete it. The contribution for this point in the objective function is at least  $\frac{k}{2}C$ , while deleting it adds at most  $\frac{k}{2}C$  to the objective function.

So, every barycenter point should be matched with at least  $\lceil \frac{k+1}{2} \rceil$  points. The total number of points is  $\sum_{j=1}^k n_j$ . Therefore, the number of barycenter points is bounded above by  $\tilde{n} = \lfloor \frac{2}{k+1} \sum_{j=1}^k n_j \rfloor$ .

It is thus sufficient to fill up all the point patterns  $\xi_j$  to  $n$  points and work also with an ansatz of  $n$  points for  $\zeta$ . Theorem 1 yields

$$\begin{aligned} & \min_{\zeta \in \mathfrak{N}_{\tilde{n}}} \sum_{j=1}^k \tau(\xi_j, \zeta)^p \\ &= \min_{z_1, \dots, z_n \in \mathcal{X}'} \sum_{j=1}^k \min_{\pi \in S_n} \sum_{i=1}^n d'(x_{\pi(i),j}, z_i)^p \\ &= \min_{z_1, \dots, z_n \in \mathcal{X}'} \min_{\pi_1, \dots, \pi_k \in S_n} \sum_{j=1}^k \sum_{i=1}^n d'(x_{\pi_j(i),j}, z_i)^p \\ &= \min_{\pi_1, \dots, \pi_k \in S_n} \sum_{i=1}^n \min_{z_i \in \mathcal{X}'} \sum_{j=1}^k d'(x_{\pi_j(i),j}, z_i)^p \end{aligned} \tag{12}$$

and that any minimizer  $\zeta_* | \mathcal{X} = \sum_{i=1}^n \delta_{z_i} | \mathcal{X}$  on the left hand side is obtained from jointly minimizing in  $\pi_1, \dots, \pi_k$  and  $z_1, \dots, z_n$  on the right hand side.  $\square$

### 4 Alternating clustering algorithms

Based on Theorem 4, we propose an algorithm that alternates between minimizing

$$\sum_{j=1}^k \sum_{i=1}^n d'(x_{\pi_j(i),j}, z_i)^p \tag{13}$$

in  $\pi_1, \dots, \pi_k \in S_n$  and in  $z_1, \dots, z_n \in \mathcal{X}'$  until convergence. Such an algorithm terminates in a local minimum of (13) after a finite number of steps, because (13) can never increase and the minimization in the permutations is over a finite space.

Since this underlying idea is close to the popular  $k$ -means clustering algorithm, we named the main function in the pseudocode and in the actual implementation `kMeansBary` (note, however, that  $n$  plays the role of  $k$  in our notation). Similar alternating algorithms in the context of Wasserstein-2 barycenters for finitely supported probability measures have been proposed in Cuturi and Doucet (2014), Borgwardt (2019) and del Barrio et al. (2019). See Sect. 5, where we compare results between `kMeansBary` and Algorithm 2 in Cuturi–Doucet.

In what follows, we present pseudocode along with the underlying ideas and explanations for two versions of the `kMeansBary`-algorithm that we dub *original* and *improved*. Here, “improved” refers to the fact that we cut down on certain computation steps in order to save runtime. We will see in Sect. 5 that this comes essentially without any performance loss.

User-friendly implementations of both algorithms are publicly available in the R-package `ttbary`; see Müller and Schuhmacher (2019).

### 4.1 Our original `kMeansBary` algorithm

The pseudocode for the basic alternating strategy described above is given in Algorithm 1. We have introduced a stopping parameter  $\delta$  to allow termination before the local optimum is reached. Since we are not interested in the actual clustering, but only in the position of the centers  $z_1, \dots, z_n$ , it seems very unlikely (though possible) that the solution changes substantially once the cost decrease has become very small. What is more, such a change might be spurious due to rounding errors in the data or when we use an approximation method for optimizing in the centers. Note also that we can always set  $\delta$  to the smallest representable positive floating-point number to ensure convergence to the local optimum.

**Algorithm 1:** `kMeansBary`. Dependence on data `pplist` suppressed for simplicity.

```

Input : center an initial pseudo-barycenter;
         pplist the list of data point patterns;
          $\delta > 0$  a constant for the termination;
          $N$  the maximum number of iterations.
Output : Locally optimal pseudo-barycenter center.

1 perm, cost  $\leftarrow$  optimPerm(center);
2 for it  $\leftarrow$  1 to  $N$  do
3   costold  $\leftarrow$  cost;
4   center  $\leftarrow$  optimBary(perm, center);
5   center  $\leftarrow$  optimDelete(perm, center);
6   center  $\leftarrow$  optimAdd(perm, center);
7   perm, cost  $\leftarrow$  optimPerm(center);
8   if costold – cost  $<$   $\delta$  then break; // difference
   always nonnegative
9 end
10 return center; // warn if the loop has run out
    
```

The minimization with respect to  $\pi_1, \dots, \pi_k$  is performed by `optimPerm`. This function computes an optimal matching between the current center and each data point pattern in `pplist`, using an alternating version of the auction algorithm with  $\varepsilon$ -scaling; see Remark 1 and Bertsekas (1988) for more details. We output the `cost` of the current matching and an  $n \times k$  matrix `perm`, whose  $j$ -th column specifies the order in which the points of the  $j$ -th data pattern are matched to  $z_1, \dots, z_n$ . For greater efficiency, we save auxiliary information (price and profit vectors) and use it for initializing the auction algorithm when calling it again with the same data point pattern.

For practical purposes, we have split up the minimization with respect to  $z_1, \dots, z_n \in \mathcal{X}'$  into a function `optimBary` that optimizes the positions within  $\mathcal{X}$  and functions `optimDelete` and `optimAdd` that optimize which of the  $z_i$  to move from  $\mathcal{X}$  to  $\aleph$  and from  $\aleph$  to  $\mathcal{X}$ , respectively. We discuss details of these functions under the separate headings below.

In addition to the outputs of the various functions shown in Algorithm 1, we also keep information on the quality of each match of points up to date. We call the match of a  $z_i$  with a data point  $x_{i'j}$

- happy* if  $z_i, x_{i'j} \in \mathcal{X}$  and  $d'(z_i, x_{i'j}) < 2^{1/p}C$
- miserable* if  $z_i, x_{i'j} \in \mathcal{X}$  and  $d'(z_i, x_{i'}) = 2^{1/p}C$
- or if  $z_i = \aleph, x_{i'j} \in \mathcal{X}$
- to  $\aleph$*  if  $x_{i'j} = \aleph$ .

Note that a miserable match is worst possible in the sense that  $\text{cost}(C_i) = \sum_{x \in C_i} d'(x, z_i)^p$  for center  $z_i$  cannot increase if  $x_{i'j}$  is replaced by *any* other  $x \in \mathcal{X}'$ .

#### Details on `optimBary`

The purpose of this function is to find for each  $z_i \in \mathcal{X}$  (i.e., not currently at  $\aleph$ ) a location in  $\mathcal{X}$  that minimizes  $\text{cost}(C_i)$  for its current cluster  $C_i = \{x_{\pi_j(i),j}; 1 \leq j \leq k\}$ . This amounts to a more traditional location problem in  $\mathcal{X}$ , except that it is typically made (much) more difficult by the fact that we have to truncate distances at  $2^{1/p}C$ .

Note that any cluster points at  $\aleph$  can be ignored because they always contribute the same amount to the cluster cost, no matter where the center lies. The same is true for individual points that have a much larger distance than  $2^{1/p}C$  from the bulk of the points. However, there are countless scenarios with (groups of) points being around distance  $2^{1/p}C$  apart from one another for which the optimization of the cluster cost becomes a difficult optimization problem (piecewise smooth on a space that is fragmented in complicated ways).

As a simple heuristic that works well in cases where we do not have to cut too many distances (i.e.,  $C$  is not too small), we suggest to ignore all points that are at the maximal  $d'$ -distance  $2^{1/p}C$  from the current  $z_i$  when computing the new  $z_i$ . Note that in this way the cluster cost can never increase.

**Algorithm 2:** `optimBary`: find optimal center pattern within  $\mathcal{X}$  for given clusters  $C_i$ .

```

1 for  $i \leftarrow$  1 to  $n$  do
2   if  $z_i \in \mathcal{X}$  then //  $z_i$  is the current center
   of the  $i$ th cluster  $C_i$ 
3     happypoints  $\leftarrow$ 
   {points in  $C_i$  that are happily matched to  $z_i$ };
4     if happypoints  $\neq \emptyset$  then
5        $z_i \leftarrow$  optimClusterCenter(happypoints);
6     end // otherwise  $z_i$  is deleted in
   next call to optimDelete
7   end
8 end
9 return  $\{z_1, \dots, z_n\}$ ;
    
```

Algorithm 2 gives corresponding pseudocode. The function `optimClusterCenter` handles the location problem for the untruncated metric  $d$  on  $\mathcal{X}$ . If for example  $\mathcal{X} = \mathbb{R}^D$  equipped with the Euclidean metric and  $p = 2$ , Equation (8) implies that `optimClusterCenter` simply has to take the (coordinatewise) average of all happy points. The case  $p = 1$  can be tackled with higher computational effort by approximation via the popular Weiszfeld algorithm; see Weiszfeld (1937).

As a further instance, which we will take up in Sect. 6, we consider the situation where  $\mathcal{X}$  is a simple graph  $(V, E)$  equipped with the shortest-path distance and  $p = 1$ . It can be shown that in this case the location problem in  $\mathcal{X}$  is solved by an element  $z_i$  of  $V \cup \mathcal{C}_i$ , i.e., either a vertex of the graph or any data point, see Hakimi (1964). We therefore proceed by first computing the distance matrix between all these points, which is then used for the entire algorithm. Such shortest-path distance computations in sparse graphs with thousands of points can be performed in (at most) a few seconds by various algorithms, see Chapter 25 in Cormen et al. (2009) and the concrete timing in Sect. 6.2. It is now easy to implement the function `optimClusterCenter`. For a given set of happy points of a cluster  $\mathcal{C}_i$ , pick the corresponding columns in the distance matrix, add them up and determine the minimal entry of the resulting vector. If there are several such entries, which due to choosing  $p = 1$  can happen quite frequently, we pick one among them uniformly at random. The index of the obtained entry identifies the center point  $z_i$ .

Precomputing the distance matrix between all points of  $V \cup \mathcal{C}_i$  in the graph case has the additional advantage that no distances have to be computed in the `optimPerm` step. It is, on the other hand, the main bottleneck of the procedure and may not be feasible in situations with very large graphs and data sets. In this case, we can resort to one of the various heuristics available, such as the single and multi-hub heuristics proposed (in principle) in Bandelt et al. (1994) and Koliander et al. (2018).

**Details on `optimDelete`**

This function deletes (i.e., moves to  $\mathfrak{N}$ ) any  $z_i \in \mathcal{X}$  for which this operation decreases  $\text{cost}(\mathcal{C}_i)$ .

We denote by  $k_{\text{happy}}$ ,  $k_{\text{miser}}$  and  $k_{\mathfrak{N}}$  the numbers of data points in  $\mathcal{C}_i$  that are happy, miserable and at  $\mathfrak{N}$ , respectively. Write furthermore  $c_{\text{happy}}$  for the total cost of matching the happy points to  $z_i$ . If  $z_i$  stays in  $\mathcal{X}$ , the cluster incurs an overall total cost of

$$c_{\text{happy}} + k_{\text{miser}} \cdot 2C^p + k_{\mathfrak{N}} \cdot C^p$$

as opposed to

$$k_{\text{happy}} \cdot C^p + k_{\text{miser}} \cdot C^p$$

if we delete  $z_i$ . Subtracting  $k_{\text{miser}} \cdot C^p$  from both expressions, this leads to the deletion condition

$$k_{\text{happy}}C^p < c_{\text{happy}} + (k - k_{\text{happy}})C^p.$$

Since  $c_{\text{happy}} \geq 0$ , a sufficient condition for deletion is  $2k_{\text{happy}} < k$ . We use this as a quick pretest, which allows us to avoid computing  $c_{\text{happy}}$  sometimes. The full deletion procedure is presented in Algorithm 3.

---

**Algorithm 3:** `optimDelete`: move center points from  $\mathcal{X}$  to  $\mathfrak{N}$  if it decreases cost.

---

```

1 for i ← 1 to n do
2   if  $z_i \in \mathfrak{N}$  then
3     happypoints ←
       {points in  $\mathcal{C}_i$  that are happily matched to  $z_i$ };
4      $k_{\text{happy}} \leftarrow \#\text{happypoints}$ ;
5     if  $2 * k_{\text{happy}} < k$  then
6        $z_i \leftarrow \mathfrak{N}$ ; // shortcut deletion
7     else
8        $c_{\text{happy}} \leftarrow \sum_{x \in \text{happypoints}} d'(x, z_i)^p$ ;
9       if  $k_{\text{happy}} * C^p < c_{\text{happy}} + (k - k_{\text{happy}}) * C^p$  then
10         $z_i \leftarrow \mathfrak{N}$ ;
11      end
12    end
13 return  $\{z_1, \dots, z_n\}$ ;

```

---

**Details on `optimAdd`**

This function adds (i.e., moves to  $\mathcal{X}$ ) any  $z_i \in \mathfrak{N}$  for which it finds a way to do so that decreases  $\text{cost}(\mathcal{C}_i)$ . Pseudocode is given in Algorithm 4.

As a compromise between computational simplicity and finding a good location in  $\mathcal{X}$ , we first sample a proposal location  $\tilde{z}$  uniformly from all miserable data points (i.e., points from any cluster that are currently in a miserable match with their center). Before we consider moving  $z_i$  to  $\tilde{z}$ , we rebuild the cluster  $\mathcal{C}_i$  in such a way that this move has a better chance of being accepted.

The corresponding procedure is performed by the `optimizeCluster`-function in the pseudocode: For each data pattern,  $\xi_j$  pick the miserable point that is closest to  $\tilde{z}$  (if it has any) and exchange it with the corresponding point  $x_{\pi_j(i),j}$  that is currently in  $\mathcal{C}_i$ . Since the point coming from the other cluster was miserable before, the cost of that cluster cannot increase by this exchange. The cost of the cluster  $\mathcal{C}_i$  can increase only if it loses a point located at  $\mathfrak{N}$  in the exchange. In this case, the cost increases by  $C^p$ , which is compensated by the fact that the cost of the other cluster must decrease, either from  $2C^p$  to  $C^p$  if its center is in  $\mathcal{X}$ , or from  $C^p$  to 0 if its center is at  $\mathfrak{N}$ . Thus, the total cost remains



the same, but  $\mathcal{C}_i$  has an additional point in  $\mathcal{X}$  now, which makes the successful addition of  $z_i$  to  $\mathcal{X}$  more likely.

To further decrease the prospective cluster cost after addition, we update the proposal  $\tilde{z}$  by recentering it in its new cluster using the appropriate `optimClusterCenter`-function introduced in `optimBary` (applied to the set of points of the new cluster that are in a happy match with  $\tilde{z}$ ).

Finally, check whether the cost of the new cluster based on the updated  $\tilde{z}$  is smaller than the same cost based on  $z_i = \aleph$ , which is  $C^p$  times the number  $k_{\mathcal{X}}$  of non- $\aleph$  points in the new cluster. Set  $z_i$  to  $\tilde{z}$  if this is the case.

**Algorithm 4:** `optimAdd`: move center points from  $\aleph$  to  $\mathcal{X}$  if it decreases cost.

```

1 alephindex ← {i ∈ [n]; z_i = ∅};
2 if alephindex ≠ ∅ then
3   supply ← {data points miserably matched to some z_i};
4   for i in alephindex do
5     if supply = ∅ then break;
6     z̃ ← sample(supply, 1); // draw uniformly
       at random from supply
7     supply ← supply \ {z̃};
8     newcluster, newperm ←
       optimizeCluster(z̃, perm, i);
9     newhappypoints ←
       {pts in newcluster with happy new match to z̃};
10    if newhappypoints ≠ ∅ then
11      z̃ ← optimClusterCenter(newhappypoints);
12      k_X ← #{x ∈ newcluster; x ∈ X};
13      c_new ← ∑_{x ∈ newcluster} d'(x, z̃)^p;
14      if c_new < k_X * C^p then
15        z_i ← z̃;
16        perm ← newperm;
17        supply ← supply \ {x ∈ supply; x is happy};
18      end
19    end
20 return {z_1, ..., z_n};

```

**4.2 An improved `kMeansBary` algorithm**

For obtaining an algorithm with a reduced computational cost, we cut down on steps that are costly, but are not expected to influence the resulting local optimum in a decisive way. Since for now we treat the location problem at the cluster level (performed by `optimBary`) as very general, allowing a wide range of metric spaces  $(\mathcal{X}, d)$ , we focus here on saving computations in the functions `optimPerm`, `optimDelete`, and `optimAdd`.

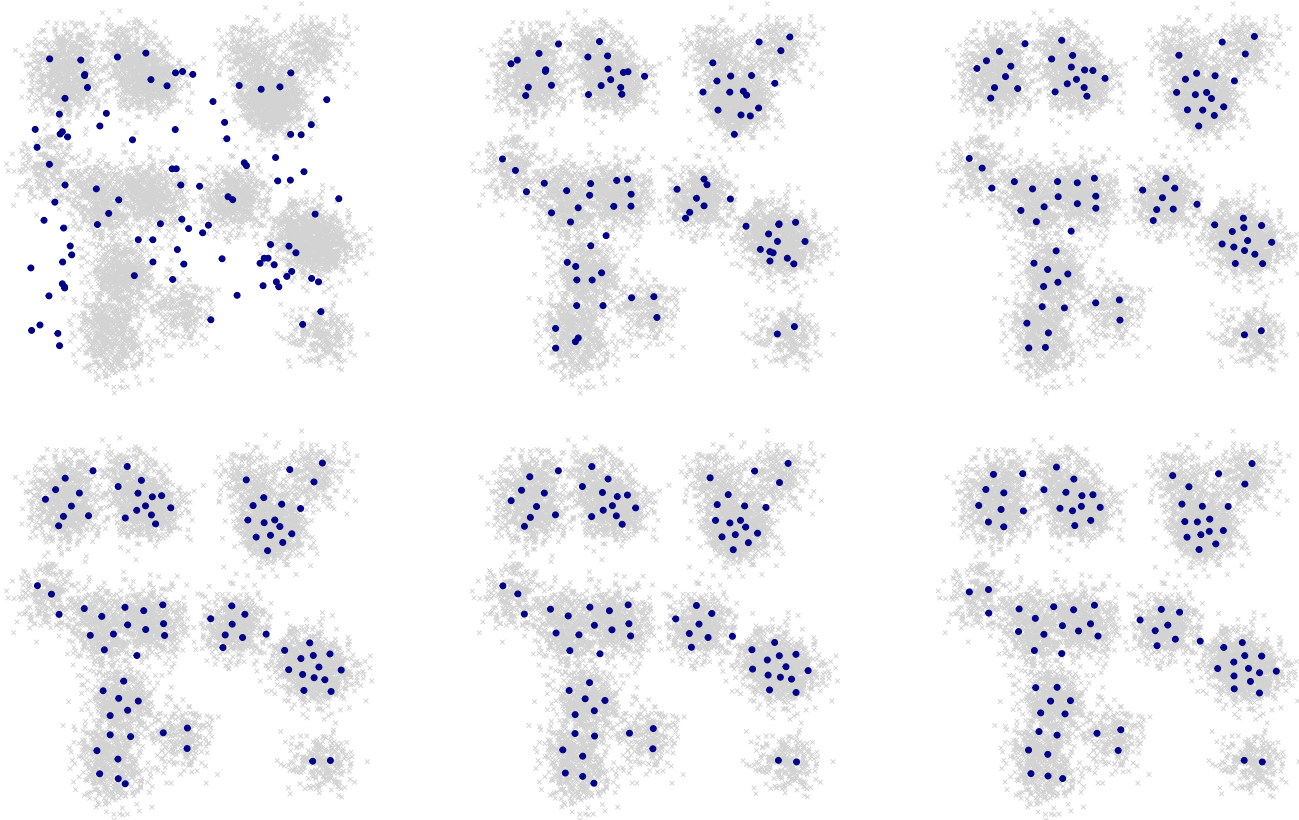
We have realized that by far the most additions and deletions of points take place in the first two iterations of the original algorithm (see also Fig. 3). Especially checking for addition of points is costly and after the first few iterations very rarely successful. Therefore, we limit such checking

henceforward to the first  $N_{\text{del/add}} = 5$  iteration steps. Some further heuristics could be applied in `optimAdd`, but the gain in computation time is not so large and they can significantly change the outcome, which is why we decided against implementing them.

In `optimPerm`, we cannot avoid doing matchings. However, the auction algorithm we use allows to solve a relaxation of the problem by stopping the  $\epsilon$ -scaling method early. In general, the auction algorithm with  $\epsilon$ -scaling based on a decreasing sequence  $(\epsilon_1, \dots, \epsilon_l)$  returns successively improved solutions that are guaranteed to lie within  $n \epsilon_i$  of the optimal total cost after the  $i$ -th step, see Bertsekas (1988, Proposition 1). By representing rescaled distances as integers in  $\{0, 1, \dots, 10^9\}$ , an optimal matching is obtained in the  $l$ -th step if  $\epsilon_l < 1/n$ . Our improved algorithm is based on the same  $\epsilon$ -vector as the original algorithm, which has components  $\epsilon_i = \frac{1}{n+1} 10^{l-i}$ ,  $1 \leq i \leq l$ , where  $l$  is chosen in such a way that  $10^7 \leq \epsilon_1 < 10^8$ . As a first improvement, we use the subsequence  $(\epsilon_{a_{it}}, \epsilon_{a_{it}+1}, \dots, \epsilon_{b_{it}})$ , where  $a$  and  $b$  are prespecified vectors of indices  $\in \{1, 2, \dots, l\}$ . A simple choice for  $a$  and  $b$  that tends to decrease the runtime noticeably is  $a_{it} = 1$  and  $b_{it} = \min\{it, l\}$ . Pseudocode for this is presented in Algorithm 5.

In practice, we settled for a somewhat more sophisticated improvement. We choose the vectors  $a = (1, 1, 1, 3, 3, 3, \dots, 3, 4)$  and  $b = (1, 2, 3, 4, 6, 8, \dots, 2 \lfloor \frac{l-1}{2} \rfloor, l)$ , and we use the sequence  $(\epsilon_{a_j}, \epsilon_{a_j+1}, \dots, \epsilon_{b_j})$ , where  $j = it$  for  $it \in \{1, 2, 3\}$ , and then  $j$  is increased by 1 each time the algorithm would otherwise converge or if the cost increases (which can only happen as long as the matchings are not optimal).

This strategy was chosen after analyzing the calculations of the algorithm with respect to the time each calculation takes. In the first two to three iterations, there are a lot of changes in the positions of the barycenter points. Especially in the first iteration, many points are deleted and added, which completely changes the assignments. Therefore, we have to begin the assignment calculation with  $\epsilon_1$  and to get more sensible results we get more precise with each of the first three iterations. After three iterations, there are usually no big changes to the barycenter anymore, so we can reuse the assignment from the iteration before as a sensible starting solution and can omit  $\epsilon_1$  and  $\epsilon_2$  in return. Leaving out the first entries of  $\epsilon$  too soon increases the runtime. Every time the algorithm converges, but has no guaranteed optimal assignment (i.e.,  $b_j < l$ ),  $j$  is increased by 1, meaning that the next two entries of  $\epsilon$  are used too, until the end of  $\epsilon$  is reached. Then, we can safely leave out the first three entries of  $\epsilon$  without increasing the runtime, because at this point the assignments from one iteration to the next only change very little.



**Fig. 3** Stepwise evolution of the barycenter for  $k = 80, m_{\#} = 100$ . In the first iteration, 32 points are deleted and 25 added. After that only movements take place

### 4.3 Practical aspects

As it turns out, the upper bound  $\tilde{n}$  on the cardinality of the barycenter from Theorem 4 is often far too large in practice. For efficiency reasons, we typically run the algorithm with a number  $n \geq \max\{n_j; 1 \leq j \leq k\}$  that is much smaller than  $\tilde{n}$ . We generate a starting point pattern center by picking  $\frac{1}{k} \sum_{j=1}^k |\xi_j|$  points uniformly at random from the underlying observation window. In a first step, all point patterns are filled to  $n$  points by adding points at  $\aleph$ . Then, Algorithm 1 or 5 is run.

Figure 3 shows a typical run of Algorithm 1 in the case of an i.i.d. sample  $\xi_1, \dots, \xi_k$  of point patterns in  $\mathbb{R}^2$  generated from a similar distribution as studied in Sect. 5. We use Euclidean distance and  $p = 2$ . The current barycenter is marked by blue points. Typically, the random starting point pattern is not a good approximation to the resulting barycenter. Therefore, many points are deleted in the first iteration. Many other ones are added at or moved to more cost efficient spots. Regardless of the starting pattern, the algorithm typically attains a reasonably looking configuration after a single iteration. After that hardly any points are added or deleted any more. The algorithm mostly moves a few individual barycenter points around each time.

### 5 Simulation study

In this section, we present a simulation study for evaluating the algorithms described in Sect. 4 for point patterns  $\xi_1, \dots, \xi_k$  in  $\mathbb{R}^2$  using squared Euclidean cost.

Unfortunately, it is not feasible for larger data examples to compute the actual barycenter as a ground truth. To illustrate this, consider the special case where all point patterns have the same cardinality  $n$  and are contained in a subset of  $\mathbb{R}^2$  of radius  $C$ . Assume further that we know that there is a barycenter that also has cardinality  $n$  (which need not be the case). In this situation, it is easy to see that instead of solving the minimization problem (13), we only need to minimize

$$\sum_{j=1}^k \sum_{i=1}^n \|x_{\pi_j(i),j} - z_i\|^2 \tag{14}$$

in  $\pi_1, \dots, \pi_k \in S_n$  and  $z_1, \dots, z_n \in \mathbb{R}^2$ . This is the assignment version of the problem of finding a barycenter of the discrete probability measures  $\frac{1}{n}\xi_1, \dots, \frac{1}{n}\xi_k$  with respect to the Wasserstein metric  $W_2$ . An exact algorithm for this problem can be found in Anderes et al. (2016) and has been tremendously improved in Borgwardt and Patterson (2018).

**Algorithm 5:** kMeansBary2. A simple version of the improved kMeansBary-algorithm. Dependence on data pplist suppressed for simplicity.

```

Input : center, pplist,  $\delta > 0$ ,  $N$  are as in the original
         kMeansBary-algorithm;
          $N_{del/add}$  number of iterations during
         which we perform delete/add steps.
Output : Locally optimal pseudo-barycenter center.
1  $l \leftarrow \lceil \log_{10}(10^8 / \frac{1}{n+1}) \rceil$ ;
2  $epsvec \leftarrow (\frac{1}{n+1} 10^{l-i})_{1 \leq i \leq l}$ ;
3  $perm, cost \leftarrow optimPerm(center, epsvec)$ ;
4 for  $it \leftarrow 1$  to  $N$  do
5    $costold \leftarrow cost$ ;
6    $center \leftarrow optimBary(perm, center)$ ;
7   if  $it \leq N_{del/add}$  then
8      $center \leftarrow optimDelete(perm, center)$ ;
9      $center \leftarrow optimAdd(perm, center)$ ;
10  end
11  if  $it < l$  then
12     $perm, cost \leftarrow optimPerm(center, epsvec[1 : it])$ ;
13  else
14     $perm, cost \leftarrow optimPerm(center, epsvec)$ ;
15    if  $costold - cost < \delta$  then break;
    // difference always nonnegative
16  end
17 end
18 return center; // warn if the loop has run out

```

Nevertheless, the computation times increase still rapidly in the problem size and reach minutes to hours for problem sizes well smaller than our smallest examples below.

Since we are not able to compare the results of our algorithm to the actual barycenter for larger examples, we assess the range of the final objective function values. In addition, we evaluate the time performance of the default algorithm and compare both objective function values and timings to the improved algorithm.

As problem instances, we created sets of  $k$  point patterns in  $\mathbb{R}^2$  having mean cardinality of  $m_{\#}$  in each pattern. The cardinalities  $n_j, j \in [k]$ , of the individual point patterns were generated by one of the following methods:

- (i) by setting  $n_j = m_{\#}$  (*deterministic cardinality*)
- (ii) by sampling  $n_j$  from a binomial distribution with mean  $m_{\#}$  and variance  $\approx 1$  (*low-variance cardinality*)
- (iii) by sampling  $n_j$  from a Poisson distribution with parameter  $m_{\#}$  (*high-variance cardinality*)

The points were distributed according to a balanced mixture of  $N \in \{5, 10, 15\}$  rotationally symmetric normal densities centered at fixed locations in  $[0, 1]^2$  and having standard deviation  $\sigma \in \{0.05, 0.1, 0.2\}$ . Figure 4 gives examples under the three center scenarios for  $k = 20$ , deterministic cardinality  $n_j = m_{\#} = 20$  and  $\sigma = 0.05$ .

We chose five  $(k, m_{\#})$  pairs  $(20, 20), (20, 50), (50, 20), (50, 50)$  and  $(100, 100)$ , which in combination with  $N$  varying in  $\{5, 10, 15\}, \sigma$  in  $\{0.05, 0.1, 0.2\}$  and the three cardinality distributions yield a total of  $5 \times 3^3 = 135$  scenarios. We created 100 instances for each scenario.

Our algorithms from Sect. 4 were run from ten starting solutions whose cardinalities matched the mean number of data points and whose points were sampled uniformly at random from  $[0, 1]^2$ . In a pilot experiment, this tended to give somewhat better local minima than starting from a random sample of all data points combined. The starting point patterns were independently chosen for each instance, but the same for both algorithms. In all cases, the penalty  $C$  was set to 0.1.

Tables 1, 2, 3 and 4 summarize the performance of our two algorithms. For the clarity of presentation, we leave

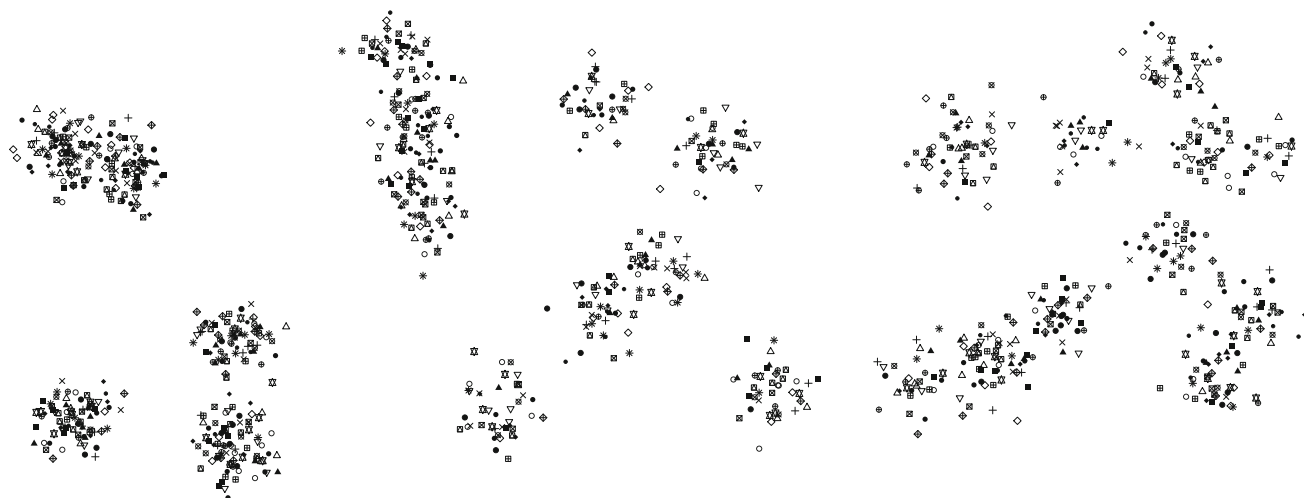


Fig. 4 20 point patterns with 20 points each from the three different center scenarios  $N = 5, 10, 15$  for  $\sigma = 0.05$

**Table 1** *Original algorithm.* Maximum relative deviations from the minimum objective function value among ten starting solutions given in percent. Means taken over 100 instances, with 0.05- and 0.95-quantiles

N	$\sigma$	20/20	20/50	50/20	50/50	100/100
5	0.05	4.59 (2.67, 7.41)	3.82 (1.93, 5.88)	4.00 (2.59, 5.93)	3.54 (2.18, 5.04)	3.27 (1.80, 4.78)
	0.2	2.06 (1.03, 3.21)	2.41 (1.42, 3.29)	1.21 (0.62, 2.03)	1.71 (1.06, 2.35)	1.32 (0.74, 1.89)
15	0.05	3.22 (1.72, 5.18)	3.31 (2.09, 5.00)	2.50 (1.25, 4.22)	2.74 (1.78, 3.89)	2.66 (1.56, 3.72)
	0.2	1.54 (0.56, 2.50)	2.19 (1.28, 3.11)	0.62 (0.01, 1.21)	1.50 (0.91, 2.21)	0.99 (0.52, 1.43)
5	0.05	3.48 (1.78, 5.46)	2.35 (1.40, 3.91)	2.82 (1.34, 4.50)	1.86 (1.04, 3.15)	1.16 (0.63, 1.81)
	0.2	1.81 (1.02, 3.01)	2.41 (1.46, 3.41)	0.97 (0.56, 1.49)	1.54 (0.97, 2.16)	1.02 (0.56, 1.66)
15	0.05	2.73 (1.48, 4.28)	2.52 (1.59, 3.89)	2.25 (1.22, 3.59)	2.04 (1.18, 3.01)	1.41 (0.79, 2.14)
	0.2	1.16 (0.42, 1.99)	2.13 (1.15, 3.13)	0.42 (0.00, 1.01)	1.42 (0.82, 2.03)	1.14 (0.61, 1.76)

in parentheses. The first block of four rows corresponds to the deterministic cardinality, the second block to the high-variance cardinality

**Table 2** *Improved algorithm.* Maximum relative deviations from the minimum objective function value of the original algorithm (both based on the same ten starting solutions) given in percent. Means over 100

N	$\sigma$	20/20	20/50	50/20	50/50	100/100
5	0.05	3.84 (1.93, 6.02)	3.77 (2.06, 5.76)	3.81 (2.02, 5.88)	3.80 (2.07, 5.63)	3.38 (1.88, 5.13)
	0.2	1.66 (0.88, 2.69)	2.49 (1.67, 3.40)	1.01 (0.45, 1.70)	1.77 (1.13, 2.42)	1.35 (0.84, 1.92)
15	0.05	3.23 (1.89, 4.81)	3.41 (2.24, 4.89)	2.61 (1.39, 4.04)	2.79 (1.67, 4.07)	2.64 (1.62, 3.82)
	0.2	1.04 (0.37, 1.81)	2.21 (1.33, 3.35)	0.40 (0.01, 0.90)	1.59 (0.99, 2.32)	0.99 (0.55, 1.41)
5	0.05	3.32 (1.80, 5.27)	2.06 (1.09, 3.12)	2.55 (1.36, 4.01)	1.67 (0.85, 2.68)	1.08 (0.49, 1.74)
	0.2	1.53 (0.63, 2.55)	2.28 (1.32, 3.40)	0.79 (0.34, 1.33)	1.53 (0.85, 2.28)	1.02 (0.59, 1.60)
15	0.05	2.49 (1.22, 4.09)	2.50 (1.63, 3.57)	2.19 (1.11, 3.57)	2.08 (1.20, 3.17)	1.33 (0.68, 2.30)
	0.2	0.82 (0.26, 1.46)	2.26 (1.44, 3.48)	0.29 (0.00, 0.77)	1.39 (0.84, 1.96)	1.12 (0.64, 1.64)

instances, with 0.05- and 0.95-quantiles in parentheses. The first block of four rows corresponds to the deterministic cardinality, the second block to the high-variance cardinality

**Table 3** *Original algorithm.* Total times in seconds for ten runs with random starting patterns. Means over 100 instances, with 0.05- and 0.95-quantiles in parentheses. The first block of four rows corresponds to the deterministic cardinality, the second block to the high-variance cardinality

N	$\sigma$	20/20	20/50	50/20	50/50	100/100
5	0.05	0.48 (0.47, 0.50)	0.89 (0.85, 0.93)	1.02 (0.99, 1.05)	2.37 (2.24, 2.52)	20.79 (19.42, 22.32)
	0.2	0.51 (0.49, 0.53)	1.14 (1.07, 1.23)	1.05 (0.99, 1.12)	3.55 (3.26, 3.96)	37.95 (34.70, 42.22)
15	0.05	0.50 (0.49, 0.51)	0.95 (0.91, 1.00)	1.06 (1.03, 1.09)	2.60 (2.44, 2.80)	24.25 (22.51, 25.93)
	0.2	0.49 (0.47, 0.51)	1.14 (1.08, 1.22)	0.91 (0.85, 0.99)	3.51 (3.12, 3.85)	37.77 (33.86, 42.21)
5	0.05	0.58 (0.53, 0.64)	1.27 (1.11, 1.53)	1.37 (1.24, 1.55)	4.12 (3.38, 4.97)	39.66 (33.33, 46.96)
	0.2	0.59 (0.53, 0.67)	1.58 (1.37, 1.88)	1.30 (1.11, 1.47)	5.62 (4.64, 6.86)	65.86 (54.51, 77.49)
15	0.05	0.59 (0.55, 0.65)	1.32 (1.16, 1.56)	1.41 (1.26, 1.64)	4.31 (3.59, 5.22)	45.37 (38.88, 51.93)
	0.2	0.55 (0.49, 0.61)	1.55 (1.37, 1.81)	1.06 (0.94, 1.23)	5.73 (4.72, 7.23)	66.25 (56.73, 79.16)

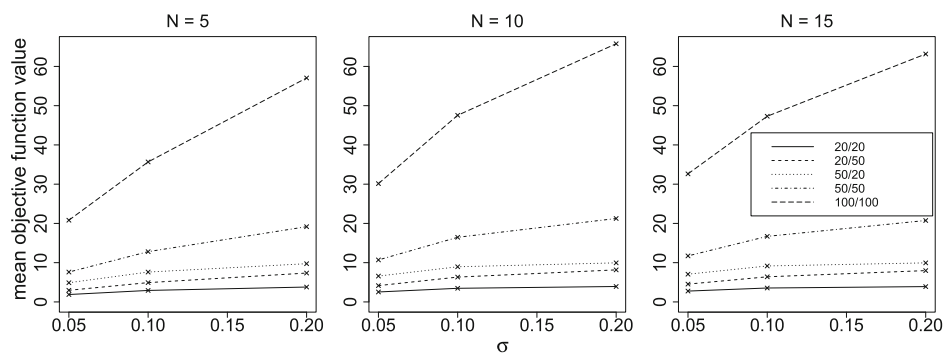
out the “middle values”  $N = 10$  and  $\sigma = 0.1$ , as well as the low-variance cardinality distribution. The corresponding performance results lie up to minor random fluctuations between the values shown. The original purpose of including the low-variance cardinality case was to detect whether a slight departure from equal cardinalities would cause substantial differences in the performance. As it turned out, this was not the case.

We first consider the original algorithm presented in Sect. 4. Table 1 gives the maximum relative deviation from the minimum  $d_{\min}$  of the resulting objective function values among the ten starting solutions, i.e.,  $\frac{d_{\max} - d_{\min}}{d_{\min}}$ . We can see that the maximal objective function value among the ten runs rarely exceeds the minimum value by more than 5%. This percentage is rather higher for the deterministic and low-variance cardinalities and when clusters in the (unmarked) superposition of the point patterns are well separated (small

**Table 4** Improved algorithm. Total times in seconds for ten runs with random starting patterns. Means over 100 instances, with 0.05- and 0.95-quantiles in parentheses. The first block of four rows corresponds to the deterministic cardinality, the second block to the high-variance cardinality

N	$\sigma$	20/20	20/50	50/20	50/50	100/100
5	0.05	0.48 (0.47, 0.48)	0.83 (0.81, 0.85)	0.97 (0.96, 0.99)	2.05 (1.99, 2.12)	13.68 (13.02, 14.57)
	0.2	0.50 (0.49, 0.50)	0.93 (0.89, 0.98)	1.00 (0.98, 1.02)	2.47 (2.33, 2.65)	18.25 (16.87, 20.57)
15	0.05	0.49 (0.48, 0.49)	0.85 (0.82, 0.88)	0.98 (0.97, 1.00)	2.08 (1.99, 2.20)	13.24 (12.47, 14.26)
	0.2	0.50 (0.49, 0.50)	0.92 (0.89, 0.95)	0.96 (0.92, 0.99)	2.42 (2.27, 2.57)	17.62 (16.08, 19.33)
5	0.05	0.56 (0.52, 0.61)	1.14 (1.00, 1.33)	1.24 (1.14, 1.36)	3.21 (2.82, 3.73)	22.47 (19.86, 25.53)
	0.2	0.58 (0.54, 0.63)	1.23 (1.09, 1.43)	1.21 (1.12, 1.30)	3.64 (3.25, 4.18)	29.41 (25.54, 34.47)
15	0.05	0.56 (0.52, 0.60)	1.12 (0.99, 1.27)	1.24 (1.13, 1.37)	3.17 (2.76, 3.71)	22.64 (19.72, 25.59)
	0.2	0.56 (0.53, 0.61)	1.19 (1.05, 1.36)	1.14 (1.04, 1.27)	3.60 (3.06, 4.35)	29.11 (24.94, 34.51)

**Fig. 5** Mean objective function values over all instances as function of  $\sigma$  for different  $N$



$N$  and  $\sigma$ ). This may well be explicable by the fact that typically many pairs can be matched over short distances in these situations such that wrong clustering decisions come typically at a higher relative cost. Figure 5 supports this by showing that the total objective function values within each problem size are lower for well separated clusters.

A further smaller experiment following up on the scenarios that exhibited the poorest performance for ten starting patterns showed that the margin of 5% increases to 8% when basing the maximum relative deviation from the minimum on 100 starting patterns.

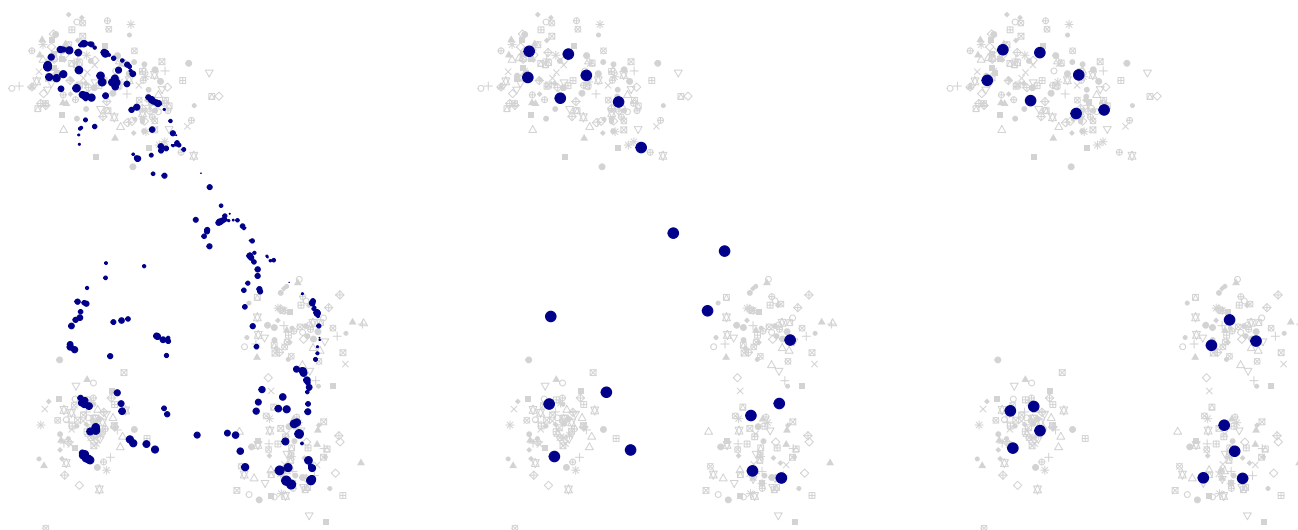
For the improved algorithm from Sect. 4.2, we compute the maximum relative deviation of its objective function values from the minimum  $d_{\min}$  of the corresponding values of the original algorithm, i.e.,  $\frac{d_{\max}^* - d_{\min}}{d_{\min}}$ , where  $d_{\max}^*$  is the maximum of the objective function values of the improved algorithm. As seen in Table 2, the performance is no worse than for the original algorithm in spite of the reduced amount of computations performed.

We finally turn to the computation times. We present the total runtimes in seconds for the ten runs with different starting patterns. This corresponds to the realistic situation of selecting as (pseudo-)barycenter the solution with the smallest local minimum in ten runs. It also provides some more stability for the means and quantiles given in Tables 3 and 4.

Table 3 gives the runtimes for the original algorithm. We see that individual runs of as large scenarios as 100 patterns with 100 points on average only take a few seconds.

From Table 4, we see that the runtimes for the improved algorithm are even considerably lower, and for some of the larger problems they have less than half of the original runtimes (at virtually no loss with regard to the objective function value as we have seen before). It is to be expected that this ratio becomes even smaller if the problem size is further increased.

Let us finally compare our algorithm to an algorithm that treats point patterns as empirical measures and tackles the Wasserstein-2 barycenter problem for these measures. As noted in the introduction, it is not realistic to treat even our smallest examples with exact algorithms for this problem. A selection of approximate algorithms can be found in Peyré and Cuturi (2019). See also the alternating algorithm in Borgwardt (2019), which includes a factor-2 performance guarantee. For our comparison, we choose Algorithm 2 in Cuturi and Doucet (2014), which alternates between solving transport problems and using gradient descent to calculate a discrete barycenter with a prescribed maximal number of support points. It allows to restrict the set of weights for the support points to a closed convex set  $\Theta$  and thus provides an approximate solution of the problem (14) if we set  $\Theta = \{(1/n, \dots, 1/n)\}$ .



**Fig. 6** Barycenters for one of our simulated data sets (20 patterns with 20 points each). From left to right: Cuturi–Doucet algorithm without constraints, Cuturi–Doucet algorithm with (maximally) 20 support

points and equal masses, typical result from `kMeansBary` algorithm based on a single start. The areas of the disks are proportional to the masses

We thank Florian Heinemann for allowing us to use his R implementation with underlying C++ code of this algorithm. Figure 6 shows an example that compares the Cuturi–Doucet algorithm without constraint using the theoretically maximal number of support points (according to Anderes et al. (2016)), the Cuturi–Doucet algorithm with full constraint and our algorithm.

To evaluate how the algorithms perform on our objective function (13), we have run both the fully constrained Cuturi–Doucet algorithm and our `kMeansBary` algorithm (with a single starting value) on the smallest scenarios used in the simulation study. These are 900 instances of 20 patterns with exactly 20 points (deterministic cardinality) and 900 instances of 20 patterns whose cardinalities are Poisson with mean 20 (high-variance cardinality).

We report the ratio of the total TT-objective function (13) between the solution of `kMeansBary` and the Cuturi–Doucet algorithm, where again  $C = 0.1$ . For the case of deterministic cardinality, the ratio was 0.729 on average, with a minimum of 0.554 and a maximum of 0.871. For the high-variance cardinality, the results are very similar with an average of 0.732 and a minimum of 0.541 and a maximum of 0.866. So on average the objective function values attained by the point patterns returned by the Cuturi–Doucet algorithm are about 37% larger than the ones attained by `kMeansBary`. This increase is reflected in the example in Fig. 6.

At the same time, the average runtime of the Cuturi–Doucet algorithm is more than twice the runtime of `kMeansBary`. This may well be due to the fact that the former is not particularly optimized for the constrained setting we use.

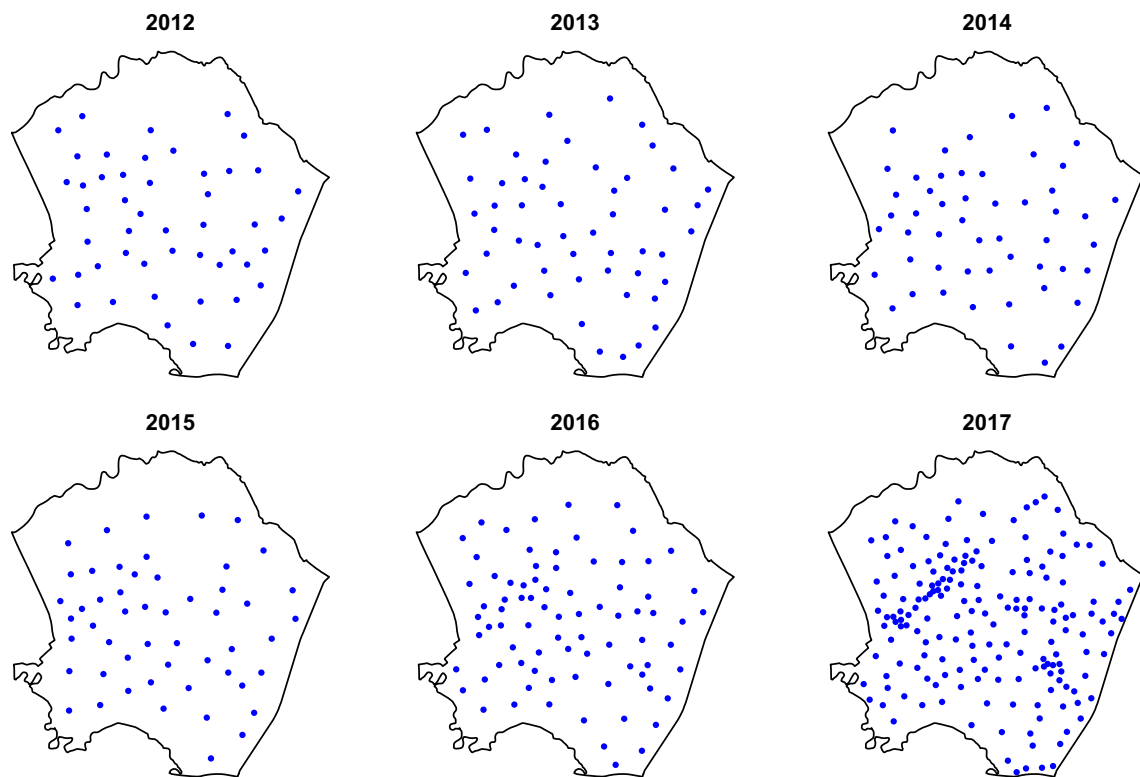
Overall, our comparison yields that the Cuturi–Doucet algorithm is not well suited for our problem, which is simply due to the fact that this algorithm was designed for a somewhat different problem. We expect similar results when comparing with other algorithms that compute (approximate) Wasserstein-2 barycenters.

## 6 Applications

The following analyses are all performed in R, see R Core Team (2019), with the help of the package `spatstat`, see Baddeley et al. (2015).

### 6.1 Street theft in Bogotá

We investigate a data set of person-related street thefts in Bogotá, Colombia, during the years 2012–2017. This data set is part of a huge data set based on a large number of types of crimes collected by the Dirección de Investigación Criminal e Interpol (DIJIN), a department of the Colombian National Police. We acknowledge DIJIN and the General Santander National Police Academy (ECSAN) for allowing us to use this data. In particular, the cases of street theft in Bogotá consist of muggings, which involve the use of force or threat, as well as pickpocketing. They do not include theft of vehicles, breaking into cars, etc. Here, we focus on the locality of Kennedy, a roughly  $7.5 \text{ km} \times 7.5 \text{ km}$  administrative ward in the west of the city, because this area is considered by the police as being more dangerous with a higher average number of crime events compared to the rest of Bogotá. The



**Fig. 7** Barycenters of weekly street thefts in the localidad of Kennedy in Bogotá. The cardinalities are 48, 53, 52, 52, 80 and 175, respectively

total number of street thefts in Kennedy for the considered period is 25 840.

Since a plot of weekly numbers of crimes reveals no clear seasonal pattern and since weekly patterns (and hence their barycenters) are of a good size to be interpreted graphically, we compute yearly barycenters for these weekly patterns. Thus, we may think of a barycenter pattern as representing a “typical week” of street thefts in the corresponding year. As penalty parameter, we chose 1000 m. Since street information was not directly available to us, we chose Euclidean distance as a metric and set  $p = 2$  to be able to relate to our simulation results in the previous section.

Each barycenter was computed based on 100 starting patterns with cardinalities regularly scattered over the integer numbers between the 0.45 to the 0.7 quantiles of the weekly number of data points for the corresponding year. We chose this somewhat asymmetrically around the median, because the mean number of thefts (the theoretical number of points in the barycenter if the penalty becomes large) was typically quite a bit to the right of the median, and also because our algorithm is somewhat better at deleting than at adding points.

Figure 7 depicts the obtained barycenters, which except for the last pattern have cardinalities just slightly below the average weekly numbers of muggings of 51.7, 57.6, 52.8, 54.4, 82.5 and 196.9, respectively. The barycenters for the

years 2012–2015 seem to be largely similar. Then, in 2016 we start seeing patterns of denser structures forming along a line to the west and a center in the south-east of Kennedy. These can be actually identified as a main street and a major intersection in the densely populated parts of Kennedy.

### 6.2 Assault cases in Valencia

As a second application, we analyze cases of assault in Valencia, Spain, reported to the police in the years 2010–2017. Since the addresses of the assaults and the street network are available, we treat this data as point patterns on a graph using shortest-path distance and  $p = 1$ . We acknowledge the local police in Valencia city together with the 112 emergency phone that kindly provided us the data after cleaning and removing any personal information.

We split up the graph and analyze the four central districts of Ciutat Vella, Eixample, Extramurs and El Pla del Real separately. For this, we assigned each assault case to its district, but added also streets from other districts at the boundary, in order to enable more natural shortest-path computations. The north-south and east-west extensions of the districts vary roughly between 1.6 and 3.3 km.

In the time domain, we split up the assault data by year and season into seven winter patterns (data from December, January and February) and eight summer patterns (data from





Considering the barycenters in Fig. 8, it seems that there are no very clear effects of the season on the assaults. Nevertheless, we may discern a number of differences between summer and winter in the four districts, which even in this relatively small data set would be considerably harder to spot in a plot of the raw data.

In the first district (Ciutat Vella), there are substantially more assaults in summer, but their spatial distribution in the barycenter is more or less similar. In Eixample, we see a concentration of assault cases in summer in the Barrio Ruzafa in the southern half of the district, whereas cases are more or less equally spread in winter. A notable feature is the occurrence of 23 barycenter points at a single crossing in summer and 7 points at the same crossing in winter with further points close by. This is due to the cul-de-sac visible in Fig. 8, which in reality forms sort of a backyard that makes the area easy for assaults, especially in summer time when there are more people (especially tourists) moving around those parts of the city. The spot is well-known to the police and in recent years the number of assaults has decreased due to police interventions. The barycenters clearly reflect this (former) assault hot spot.

In the district of Extramurs, both barycenters are more or less spread over the whole district, with two clusters of assaults occurring in the east and south. Both clusters are much more pronounced in winter. In the district of El Pla del Real, there is some concentration in the winter month in the east and south-east. Apart from that the only noticeable difference is that there are substantially more assaults in winter than in summer, which may well be related to the fact that this is a popular student district.

### 7 Discussion and outlook

In this paper, we have introduced the  $p$ -th-order TT- and RTT-metrics, which allow us to measure distances between point patterns in an intuitive way, generalizing several earlier metrics. We have investigated  $q$ -th-order barycenters with respect to the TT-metric and presented two variants of a heuristic algorithm. These variants return local minimizers of the Fréchet functional that mimic properties of the actual barycenter well and attain consistent objective function values. They are computable in a few seconds for medium-sized problems, such as 100 patterns of 100 points.

For the proof of Theorem 4, it was necessary to set  $p = q$ . While such a choice may seem natural, we point out that due to the separate interpretations of  $p$  as the order for matching points in the metric on  $\mathfrak{N}_{\text{fin}}$  (higher  $p$  tends to balance out the matching distances) and  $q$  as the order of the empirical moment in  $\mathfrak{N}_{\text{fin}}$ , it may well be desirable to combine  $p \neq q$ .

In the present paper, we have only dealt with the descriptive aspects of barycenters. It is thus clear that our appli-

cations in Sect. 6 can only be seen as explorative studies. In order to determine whether differences between group barycenters are statistically significant, we need to take the distribution of the point patterns around their barycenters into account and perform appropriate hypothesis tests.

Fortunately, the Fréchet functional (7) provides us with a natural quantification of scatter around the barycenter. For  $q = 2$ , it is quite common to refer to

$$\text{Var}(\xi_1, \dots, \xi_k) = \min_{\zeta \in \mathfrak{N}_{\text{fin}}} \frac{1}{k} \sum_{j=1}^k \tau(\xi_j, \zeta)^2$$

as (empirical) Fréchet variance, due to Equation (8). Detailed asymptotic theory for performing an analysis of variance (ANOVA) in metric spaces based on comparing Fréchet variances has been recently developed in Dubey and Müller (2019a). The application and adaptation of this theory for the point pattern space and an investigation of the performance of our heuristic algorithm in this context will be the subject of a future paper.

Based on the computation of barycenters further more advanced procedures in statistics and machine learning become possible. This includes barycenter-based dimension reduction techniques, such as Wasserstein dictionary learning, see Schmitz et al. (2018), and functional principal component analysis of point patterns evolving in time, see Dubey and Müller (2019b).

**Acknowledgements** Open Access funding provided by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### Appendix: Proofs left out in the main text

**Lemma A.1** *Let  $C > 0$ ,  $\tilde{C} \in (0, 2C]$  and let  $(\mathcal{X}, d)$  be a metric space with  $\text{diam}(\mathcal{X}) = \sup_{x, y \in \mathcal{X}} d(x, y) \leq 2C$ . For  $k \in \mathbb{N}$  set  $\mathcal{X}' = \mathcal{X} \cup \{\mathfrak{N}_1, \dots, \mathfrak{N}_k\}$ , where  $\mathfrak{N}_1, \dots, \mathfrak{N}_k \notin \mathcal{X}$  are pairwise different, and define*

$$d'(x, y) = \begin{cases} d(x, y) & \text{if } x, y \in \mathcal{X}; \\ C & \text{if } \{x, y\} \cap \mathcal{X} \neq \emptyset \\ & \text{and } \{x, y\} \cap \{\mathfrak{N}_1, \dots, \mathfrak{N}_k\} \neq \emptyset; \\ \tilde{C} & \text{if } \{x, y\} \subset \{\mathfrak{N}_1, \dots, \mathfrak{N}_k\} \text{ and } x \neq y; \\ 0 & \text{if } x = y = \mathfrak{N}_i \text{ for some } i \in [k]. \end{cases}$$

Then,  $(\mathcal{X}', d')$  is a metric space.

**Proof** Identity and symmetry properties of the map  $d' : \mathcal{X}' \times \mathcal{X}' \rightarrow \mathbb{R}_+$  follow immediately. Since  $d$  is a metric on  $\mathcal{X}$  and  $\tilde{d}(x, y) = \tilde{C}$  if  $x \neq y$  (and 0 otherwise) defines a metric  $\tilde{d}$  on  $\mathcal{Y} = \{\aleph_1, \dots, \aleph_k\}$ , we only have to check the triangle inequality for a few special cases. If  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$  (or vice versa), then  $d(x, y) = C$ . Since one of  $d(x, z)$  and  $d(z, y)$  has to be  $= C$  regardless of  $z \in \mathcal{X}'$ , we obtain  $d(x, y) \leq d(x, z) + d(z, y)$ . If  $x, y \in \mathcal{X}$  and  $z \in \mathcal{Y}$ , then

$$d(x, y) \leq \text{diam}(\mathcal{X}) \leq 2C = d(x, z) + d(z, y).$$

Likewise, if  $x, y \in \mathcal{Y}$  and  $z \in \mathcal{X}$ , then

$$d(x, y) \leq \tilde{C} \leq 2C = d(x, z) + d(z, y).$$

□

**Proof of Theorem 1** Since  $\bar{\tau}(\xi, \eta) = \frac{1}{n^{1/p}} \tau(\xi, \eta)$ , it is enough to show the statement for  $\tau$ .

Let  $\pi \in S_n$  be a permutation that minimizes  $\sum_{i=1}^n d'(x_i, y_{\pi(i)})^p$ . Writing  $I = \{i \in [m]; d(x_i, y_{\pi(i)}) < 2^{1/p}C\}$ , we obtain

$$d'(x_i, y_{\pi(i)}) = \begin{cases} d(x_i, y_{\pi(i)}) & \text{if } i \in I; \\ 2^{1/p}C & \text{if } i \in [m] \setminus I; \\ C & \text{if } i \in [n] \setminus [m]. \end{cases} \quad (\text{A.15})$$

Therefore, enumerating  $I$  in arbitrary order as  $\{i_1, \dots, i_l\}$  for some  $l \in [m]$  and setting  $j_r := \pi(i_r)$  for  $r \in [l]$ , we have

$$\begin{aligned} & \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p \\ &= \sum_{r=1}^l d(x_{i_r}, y_{j_r})^p + (m-l)(2^{1/p}C)^p + (n-m)C^p \\ &= (m+n-2l)C^p + \sum_{r=1}^l d(x_{i_r}, y_{j_r})^p. \end{aligned} \quad (\text{A.16})$$

Thus,  $\tau(\xi, \eta)^p \leq \min_{\pi \in S_n} \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p$ .

Conversely, let  $(i_1, \dots, i_l; j_1, \dots, j_l) \in S(m, n)$  minimize  $(m+n-2l)C^p + \sum_{r=1}^l d(x_{i_r}, y_{j_r})^p$ . This implies  $d(x_{i_r}, y_{j_r}) \leq 2^{1/p}C$  for all  $r \in [l]$ , because otherwise we could obtain a smaller value by removing  $i_r, j_r$  from the vector. Writing  $I = \{i_1, \dots, i_l\}$ ,  $J = \{j_1, \dots, j_l\}$  it implies also that  $d(x_i, y_j) \geq 2^{1/p}C$  for all  $i \in [m] \setminus I$  and  $j \in [n] \setminus J$ , because otherwise we could obtain a smaller value by adding  $i, j$  to the vector. Let then  $\pi \in S_n$  be any permutation satisfying  $\pi(i_r) = \pi(j_r)$  for all  $r \in [l]$ . With this  $\pi$ , we obtain exactly the  $d'$ -distances in (A.15) for all  $i \in [n]$  and hence (A.16) holds again. Thus,  $\min_{\pi \in S_n} \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p \leq \tau(\xi, \eta)^p$ . □

**Proof of Theorem 2** We start with the map  $\tau : \mathfrak{N}_{\text{fin}} \times \mathfrak{N}_{\text{fin}} \rightarrow \mathbb{R}_+$ . If  $\xi = \eta$ , then  $m = n$  and there is a permutation  $\pi \in S_n$  such that  $x_i = y_{\pi(i)}$  for  $1 \leq i \leq n$ . Hence  $\tau(\xi, \eta) = 0$ , choosing  $l = n$  and  $(i_1, \dots, i_l; j_1, \dots, j_l) = (1, \dots, n, \pi(1), \dots, \pi(n))$ . If on the other hand  $\tau(\xi, \eta) = 0$ , we must have  $l = m = n$  to be able to achieve  $m+n-2l = 0$  and there must be  $(i_1, \dots, i_n; j_1, \dots, j_n) \in S(n, n)$  such that  $d(x_{i_r}, y_{j_r}) = 0$  for  $1 \leq r \leq n$ . Since the  $d$  is a metric, this yields  $\xi = \sum_{r=1}^n \delta_{i_r} = \sum_{r=1}^n \delta_{j_r} = \eta$ . The symmetry of  $\tau$  is immediately clear from the symmetric form of (2).

For the proof of the triangle inequality, we use the metric space  $(\mathcal{X}', d')$  introduced before Theorem 1. Let  $\xi, \eta, \zeta \in \mathfrak{N}_{\text{fin}}$ . After filling up patterns to the maximum of the three cardinalities by adding points at the auxiliary location  $\aleph$ , we may assume that  $\xi = \sum_{i=1}^n \delta_{x_i}$ ,  $\eta = \sum_{j=1}^n \delta_{y_j}$  and  $\zeta = \sum_{k=1}^n \delta_{z_k}$  have the same cardinality. Noting that given two point patterns of the same cardinality we may add any number of extra points located at  $\aleph$  to both of them without changing their  $\tau$ -distance, Theorem 1 yields that there are  $\pi_1, \pi_2 \in S_n$  such that

$$\begin{aligned} \tau(\xi, \zeta) &= \left( \sum_{i=1}^n d'(x_i, z_{\pi_1(i)})^p \right)^{1/p} \quad \text{and} \\ \tau(\zeta, \eta) &= \left( \sum_{i=1}^n d'(z_i, y_{\pi_2(i)})^p \right)^{1/p}. \end{aligned}$$

Then,  $\pi = \pi_2 \circ \pi_1 \in S_n$  matches the points of  $\xi$  and  $\eta$  in such a way that

$$d'(x_i, y_{\pi(i)}) \leq d'(x_i, z_{\pi_1(i)}) + d'(z_{\pi_1(i)}, y_{\pi_2(\pi_1(i))})$$

and Theorem 1 and the triangle inequality for the  $\ell_p$ -norm yields that

$$\begin{aligned} \tau(\xi, \eta) &\leq \left( \sum_{i=1}^n d'(x_i, y_{\pi(i)})^p \right)^{1/p} \\ &\leq \left( \sum_{i=1}^n d'(x_i, z_{\pi_1(i)})^p \right)^{1/p} \\ &\quad + \left( \sum_{i=1}^n d'(z_{\pi_1(i)}, y_{\pi_2(\pi_1(i))})^p \right)^{1/p} \\ &= \tau(\xi, \zeta) + \tau(\zeta, \eta). \end{aligned}$$

We turn to the map  $\bar{\tau} : \mathfrak{N}_{\text{fin}} \times \mathfrak{N}_{\text{fin}} \rightarrow \mathbb{R}_+$ . Since  $\bar{\tau}(\xi, \eta) = \frac{1}{\max\{|\xi|, |\eta|\}^{1/p}} \tau(\xi, \eta)$ , we may inherit the identity and symmetry properties for  $\bar{\tau}$  directly from  $\tau$ . To show the triangle inequality, let  $\xi = \sum_{i=1}^{m_1} \delta_{x_i}$ ,  $\eta = \sum_{j=1}^{m_2} \delta_{y_j}$  and  $\zeta = \sum_{k=1}^{m_*} \delta_{z_k}$  be in  $\mathfrak{N}_{\text{fin}}$  and set  $m_* = \max\{m_1, m_2\}$ . If  $n \leq m_*$ , we obtain the desired result from the triangle

inequality of  $\tau$  as

$$\begin{aligned} \bar{\tau}(\xi, \eta) &= \frac{1}{m_*^{1/p}} \tau(\xi, \eta) \\ &\leq \frac{1}{m_*^{1/p}} (\tau(\xi, \zeta) + \tau(\zeta, \eta)) \\ &\leq \frac{1}{\max\{m_1, n\}^{1/p}} \tau(\xi, \zeta) + \frac{1}{\max\{n, m_2\}^{1/p}} \tau(\zeta, \eta) \\ &\leq \bar{\tau}(\xi, \zeta) + \bar{\tau}(\zeta, \eta). \end{aligned}$$

If  $n > m_*$ , we use a slightly different construction for the extended metric space. Let  $\mathcal{X}' = \mathcal{X} \cup \{\aleph, \aleph'\}$  for two different  $\aleph, \aleph' \notin \mathcal{X}$ . Setting

$$d'(x, y) = \begin{cases} \min\{d(x, y), 2^{1/p}C\} & \text{if } x, y \in \mathcal{X}, \\ C & \text{if } \{x, y\} \cap \mathcal{X} \neq \emptyset \text{ and} \\ & \{x, y\} \cap \{\aleph, \aleph'\} \neq \emptyset, \\ 2^{1/p}C & \text{if } \{x, y\} = \{\aleph, \aleph'\}, \\ 0 & \text{if } x = y = \aleph \\ & \text{or } x = y = \aleph', \end{cases}$$

we obtain by Lemma A.1 that  $(\mathcal{X}', d')$  is again a metric space. Setting  $x_i = \aleph$  for  $m_1 + 1 \leq i \leq n$  and  $y_j = \aleph'$  for  $m_2 + 1 \leq j \leq n$ , we may define  $\tilde{\xi} = \sum_{i=1}^n \delta_{x_i}$  and  $\tilde{\eta} = \sum_{j=1}^n \delta_{y_j}$ . Note that an optimal permutation  $\pi_* \in S_{m_*}$  for  $\bar{\tau}(\xi, \eta)$  can be extended to an optimal permutation  $\tilde{\pi}_* \in S_n$  for  $\bar{\tau}(\tilde{\xi}, \tilde{\eta})$  by setting  $\tilde{\pi}_*(i) = i$  for  $m_* + 1 \leq i \leq n$ . Furthermore, for any  $s, c \geq 0$  with  $s \leq m_*c$ , we have  $\frac{1}{m_*}s \leq \frac{1}{n}(s + (n - m_*)c)$ . Combining these two facts, we obtain

$$\begin{aligned} \bar{\tau}(\xi, \eta)^p &= \frac{1}{m_*} \sum_{i=1}^{m_*} d'(x_i, y_{\pi_*(i)})^p \\ &\leq \frac{1}{n} \left( \sum_{i=1}^{m_*} d'(x_i, y_{\pi_*(i)})^p + (n - m_*) \cdot 2C^p \right) \\ &= \bar{\tau}(\tilde{\xi}, \tilde{\eta})^p, \end{aligned}$$

and therefore

$$\bar{\tau}(\xi, \eta) \leq \bar{\tau}(\tilde{\xi}, \tilde{\eta}) \leq \bar{\tau}(\tilde{\xi}, \zeta) + \bar{\tau}(\zeta, \tilde{\eta}) = \bar{\tau}(\xi, \zeta) + \bar{\tau}(\zeta, \eta),$$

where the second inequality holds since the cardinalities of all point patterns are equal and the equality holds by two more applications of Theorem 1.  $\square$

**Proof of Theorem 3** The equivalence for (a) was already used in Diez et al. (2012). We give a quick argument for the sake of completeness. We may assume without loss of generality that, in an admissible path  $P = (\xi_0, \dots, \xi_N)$  for the minimization problem (4),

- only moves from  $x \in \xi$  to  $y \in \eta$  occur;

- only points  $y \in \eta$  are added;
- only points  $x \in \xi$  are deleted;

because if any of these conditions were violated, the total cost of the path could only become larger (for the first item we use the triangle inequality for  $d$ ). The minimization (4) is then equivalent to choosing  $l \in \{0, 1, \dots, \min\{m, n\}\}$  points to be moved from  $\xi$ -points with indices  $i_1, \dots, i_l \in [m]$  to  $\eta$ -points with indices  $j_1, \dots, j_l \in [n]$ , respectively, at cost  $d(x_{i_r}, y_{j_r})$  for each move. The remaining  $m - l$  points of  $\xi$  are deleted at cost  $C$  per deletion, and the remaining  $n - l$  points of  $\eta$  are added at cost  $C$  per addition. This yields exactly the minimization problem (2).

The equivalence (b) is an immediate consequence of Theorem 1.  $\square$

### References

Agueh, M., Carlier, G.: Barycenters in the Wasserstein space. *SIAM J. Math. Anal.* **43**, 904–924 (2011)

Anderes, E., Borgwardt, S., Miller, J.: Discrete Wasserstein barycenters: optimal transport for discrete data. *Math. Methods Oper. Res.* **84**(2), 389–409 (2016)

Baddeley, A., Rubak, E., Turner, R.: *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC, Boca Raton (2015)

Bandelt, H.-J., Crama, Y., Spieksma, F.C.R.: Approximation algorithms for multi-dimensional assignment problems with decomposable costs. *Discrete Appl. Math.* **49**, 25–50 (1994)

Bertsekas, D.P.: The auction algorithm: a distributed relaxation method for the assignment problem. *Ann. Oper. Res.* **14**, 105–123 (1988)

Błaszczyszyn, B., Haenggi, M., Keeler, P., Mukherjee, S.: *Stochastic Geometry Analysis of Cellular Networks*. Cambridge University Press, Cambridge (2018)

Borgwardt, S.: An LP-based, strongly polynomial 2-approximation algorithm for sparse Wasserstein barycenters. Preprint (2019). [arXiv:1704.05491v5](https://arxiv.org/abs/1704.05491v5)

Borgwardt, S., Patterson, S.: Improved linear programs for discrete barycenters. *INFORMS J Optim* (2018). [arXiv:1803.11313](https://arxiv.org/abs/1803.11313)

Chiaraviglio, L., Cuomo, F., Maisto, M., Gigli, A., Lorincz, J., Zhou, Y., Zhao, Z., Qi, C., Zhang, H.: What is the best spatial distribution to model base station density? A deep dive into two European mobile networks. *IEEE Access* **4**, 1434–1443 (2016)

Chizat, L.: *Unbalanced Optimal Transport: Models, Numerical Methods, Applications*. Ph.D. thesis, PSL Research University (2017)

Chizat, L., Peyré, G., Schmitzer, B., Vialard, F.-X.: Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.* **87**(314), 2563–2609 (2018)

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT Press, Cambridge (2009)

Cuturi, M., Doucet, A.: Fast computation of Wasserstein barycenters. In: Xing, E.P., Jebara, T. (eds.) *Proceedings of the 31st International Conference on Machine Learning*, pp. 685–693 (2014)

del Barrio, E., Cuesta-Albertos, J.A., Matrán, C., Mayo-Íscar, A.: Robust clustering tools based on optimal transportation. *Stat. Comput.* **29**, 139–160 (2019)

Diez, D.M., Schoenberg, F.P., Woody, C.D.: Algorithms for computing spike time distance and point process prototypes with application to feline neuronal responses to acoustic stimuli. *J. Neurosci. Methods* **203**(1), 186–192 (2012)

- Diggle, P.J.: Statistical Analysis of Spatial and Spatio-Temporal Point Patterns. Chapman and Hall/CRC, Boca Raton (2013)
- Dubey, P., Müller, H.-G.: Fréchet analysis of variance for random objects. Preprint **106**(4), 803–821 (2019a)
- Dubey, P., Müller, H.-G.: Functional models for time-varying random objects. Preprint (2019b). [arXiv:1907.10829](https://arxiv.org/abs/1907.10829)
- Fréchet, M.: Les éléments aléatoires de nature quelconque dans un espace distancié. Ann. Inst. H. Poincaré **10**, 215–310 (1948)
- Hakimi, S.L.: Optimum locations of switching centers and the absolute centers and medians of a graph. Oper. Res. **12**(3), 456–458 (1964)
- Koliander, G., Schuhmacher, D., Hlawatsch, F.: Rate-distortion theory of finite point processes. IEEE Trans. Inf. Theory **64**(8), 5832–5861 (2018)
- Konstantinoudis, G., Schuhmacher, D., Ammann, R., Diesch, T., Kuehni, C., Spycher, B.D.: Bayesian spatial modelling of childhood cancer incidence in Switzerland using exact point data: a nationwide study during 1985–2015. Preprint (2019). <https://www.medrxiv.org/content/early/2019/07/15/19001545>
- Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Quart. **2**, 83–97 (1955)
- Liero, M., Mielke, A., Savaré, G.: Optimal entropy-transport problems and a new Hellinger-Kantorovich distance between positive measures. Invent. Math. **211**(3), 969–1117 (2018)
- Lin, Z., Müller, H.-G.: Total variation regularized Fréchet regression for metric-space valued data. Preprint (2019). [arXiv:1904.09647](https://arxiv.org/abs/1904.09647)
- Lombardo, L., Opitz, T., Huser, R.: Point process-based modeling of multiple debris flow landslides using INLA: an application to the 2009 Messina disaster. Stoch. Environ. Res Risk Assess. **32**(7), 2179–2198 (2018)
- Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming, third edn. Springer, New York (2008)
- Mateu, J., Schoenberg, F.P., Diez, D.M., Gonzáles, J.A., Lu, W.: On measures of dissimilarity between point patterns: classification based on prototypes and multidimensional scaling. Biom. J. **57**(2), 340–358 (2015)
- Moradi, M., Mateu, J.: First and second-order characteristics of spatio-temporal point processes on linear networks. J. Comput. Graph. Stat. (2019, to appear)
- Moradi, M., Rodriguez-Cortes, F., Mateu, J.: On kernel-based intensity estimation of spatial point patterns on linear networks. J. Comput. Graph. Stat. **27**(2), 302–311 (2018)
- Müller, R., Schuhmacher, D.: ttbary: barycenter methods for spatial point patterns. R package version 0.1-1. (2019) <https://cran.r-project.org/package=ttbary>
- Petersen, A., Müller, H.-G.: Fréchet regression for random objects with Euclidean predictors. Ann. Stat. **47**(2), 691–719 (2019)
- Peyré, G., Cuturi, M.: Computational optimal transport Foundations and Trends®. Mach. Learn. **11**(5–6), 355–607 (2019)
- R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2019)
- Rakshit, S., Davies, T., Moradi, M., McSwiggan, G., Nair, G., Mateu, J., Baddeley, A.: Fast kernel smoothing of point patterns on a large network using 2d convolution. Int. Stat. Rev. (2019)
- Samartsidis, P., Eickhoff, C.R., Eickhoff, S.B., Wager, T.D., Barrett, L.F., Atzil, S., Johnson, T.D., Nichols, T.E.: Bayesian log-Gaussian Cox process regression: applications to meta-analysis of neuroimaging working memory studies. J. R. Stat. Soc. Ser. C **68**(1), 217–234 (2019)
- Schmitz, M.A., Heitz, M., Bonneel, N., Ngole, F., Coeurjolly, D., Cuturi, M., Peyré, G., Starck, J.-L.: Wasserstein dictionary learning: optimal transport-based unsupervised nonlinear dictionary learning. SIAM J. Imaging Sci. **11**(1), 643–678 (2018)
- Schoenberg, F.P., Tranbarger, K.E.: Description of earthquake after-shock sequences using prototype point patterns. Environmetrics **19**(3), 271–286 (2008)
- Schuhmacher, D.: Stein’s method for approximating complex distributions, with a view towards point processes. In: Schmidt, V. (ed.) Stochastic Geometry, Spatial Statistics and Random Fields, Vol. II: Models and Algorithms. Lecture Notes in Mathematics, vol. 2120, pp. 1–30. Springer (2014)
- Schuhmacher, D., Vo, B.-T., Vo, B.-N.: A consistent metric for performance evaluation of multi-object filters. IEEE Trans. Signal Process. **56**(8, part 1), 3447–3457 (2008)
- Schuhmacher, D., Xia, A.: A new metric between distributions of point processes. Adv. Appl. Probab. **40**(3), 651–672 (2008)
- Victor, J.D., Purpura, K.P.: Metric-space analysis of spike trains: theory, algorithms and application. Netw. Comput Neural Syst. **8**, 127–164 (1997)
- Weiszfeld, E.: Sur le point pour lequel la somme des distances de  $n$  points donnés est minimum. Tohoku Math. J. **43**, 355–386 (1937)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.